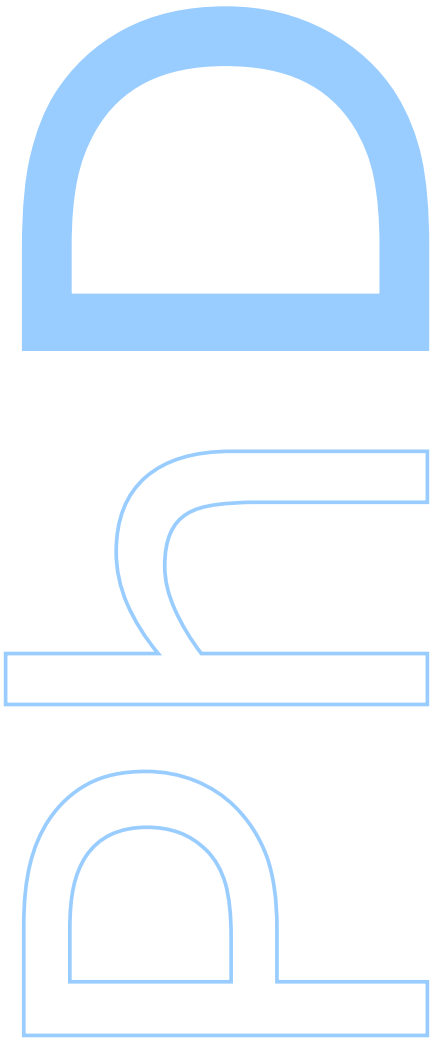
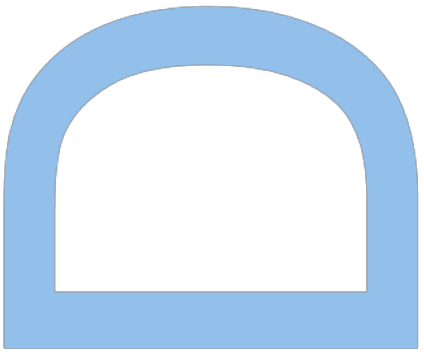
Learning Word Sense Representations from Neural Language Models

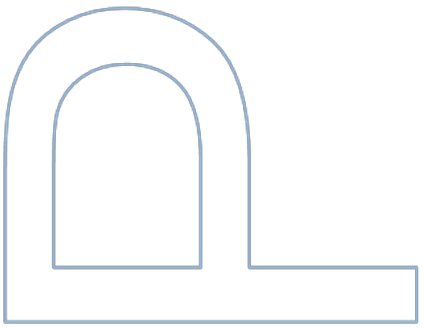
Daniel Loureiro
Programa Doutoral em Ciência de Computadores
Departamento de Ciência de Computadores
2023

Orientador
Prof. Dr. Alípio Jorge, Faculdade de Ciências da Universidade do Porto




U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO





UNIVERSIDADE DO PORTO

DOCTORAL THESIS

Learning Word Sense Representations from Neural Language Models

Author:
Daniel LOUREIRO

Supervisor:
Alípio JORGE

A thesis submitted in fulfilment of the requirements
for the degree of Ph.D. in Computer Science

at the

Faculdade de Ciências da Universidade do Porto
Departamento de Ciência de Computadores

2023

" It is change, continuing change, inevitable change, that is the dominant factor in society today. No sensible decision can be made any longer without taking into account not only the world as it is, but the world as it will be. "

Isaac Asimov (1978)

Declaração de Honra

Eu, Daniel Alexandre Bouçanova Loureiro, inscrito(a) no Programa Doutoral em Ciência de Computadores da Faculdade de Ciências da Universidade do Porto declaro, nos termos do disposto na alínea a) do artigo 14.º do Código Ético de Conduta Académica da U.Porto, que o conteúdo da presente tese reflete as perspetivas, o trabalho de investigação e as minhas interpretações no momento da sua entrega.

Ao entregar esta tese, declaro, ainda, que a mesma é resultado do meu próprio trabalho de investigação e contém contributos que não foram utilizados previamente noutros trabalhos apresentados a esta ou outra instituição.

Mais declaro que todas as referências a outros autores respeitam escrupulosamente as regras da atribuição, encontrando-se devidamente citadas no corpo do texto e identificadas na secção de referências bibliográficas. Não são divulgados na presente tese quaisquer conteúdos cuja reprodução esteja vedada por direitos de autor.

Tenho consciência de que a prática de plágio e auto-plágio constitui um ilícito académico.

Daniel Alexandre Bouçanova Loureiro

Porto, 19 de janeiro de 2023

Acknowledgements

I journeyed into this Ph.D. from a place of challenging myself on whether a life-long wonder about Artificial Intelligence could ever be materialized into any sort of contributions towards its progress, no matter how small. Pursuing this personal challenge has been a privilege made possible by numerous people I have been fortunate to have in my life, and crossed in my path along the way.

First, and foremost, I owe this achievement to my parents. My father has instilled in me a life-long passion for science, and the self-confidence to believe its understanding is within my reach. My mother broadened my worldview, and helped me develop an ownership mindset which proved crucial for this journey. Without those ingredients in my upbringing, research would not be a part of my life.

I am also sincerely grateful to have Prof. Alípio Jorge as my supervisor. He took a chance in accepting me for the Ph.D. when I had little more than an ambitious plan. Besides his guidance and constructive criticism, I am particularly grateful for the freedom he granted me to pursue my interests. I also thank Prof. Sabine Broda for helping me understand the feasibility of applying for the Ph.D., and Prof. Goreti Marreiros for introducing me to Natural Language Processing research during my bachelor's. My chance encounter with Prof. José Camacho-Collados also had an outsized impact on the course of this thesis, and ultimately my research career – I am thankful for his time and generosity.

I am deeply grateful to my wife, Vilma Ramos, for her patience and unabated support, despite the many challenges we faced during these trying times. Her intelligence and resilience are qualities I may only ever aspire to reach. I also thank my in-laws and extended family, without their support this work would not have been possible.

My dear friend and lab mate David Aparício provided many much-needed breaks from the daily grind of research. His friendship and example helped greatly. Our lunches together with Prof. Pedro Ribeiro were also always fun and invigorating. I am also thankful for the friendliness of my other fellow DCC lab mates, namely Nuno Guimarães, Jorge Silva, Arian Pasquali, and Shamsuddeen Hassan. Helping out on various projects with Prof. Nuno Moniz was also another highlight of my time at DCC.

I dedicate this thesis to my children, J. Lucas, M. Mateus, and M. Helena, who give meaning to all of this. I hope this thesis may provide them a relatable example of what is achievable when there is determination, hard-work, and a bit of good fortune too.

Abstract

The recent development of large Neural Language Models (NLMs) based on end-to-end deep learning architectures has delivered unprecedented breakthroughs in the field of Natural Language Processing (NLP).

Most remarkably, this progress has been driven by a simple pipeline that consists of initially training a NLM, without supervision, on large corpora (i.e., pre-training), followed by fine-tuning that same NLM for different tasks using relatively small task-specific datasets. Consequently, repeating this process using NLMs with additional parameters yields performance improvements across tasks.

However, this improvement from scaling data and computational resources is expected to plateau in the near future, and hybrid solutions combining NLMs with complementary approaches, such as Probabilistic Logic (PL), are being actively researched.

In this hybrid setup, NLMs are expected to produce contextual representations which complementary models can use for logical reasoning, a known limitation of NLMs. The nature of these representations, and the interface between NLMs and PL, are major open research questions in NLP.

In this thesis, we explore how to extract accurate sense-level representations from the internal states of NLMs towards the development of hybrid solutions that can leverage NLMs for symbolic representation of commonsense knowledge. Our proposed methods, based on contextual embeddings, allow for matching words and phrases within texts to the broad set of commonsense concepts covered by the popular WordNet ontology. Instead of performing task-specific fine-tuning or training other models based on features from NLMs, our approach exploits the latent spaces learned from self-supervised pre-training of NLMs, using nearest neighbors (k-NN) for inference. Additionally, we also explore how our proposed representations can improve zero-shot relation extraction from NLMs, focusing on relations relevant for commonsense reasoning (e.g., *UsedFor*).

We evaluate our progress using various tasks related to Word Sense Disambiguation (WSD), presenting state-of-the-art results on several of these tasks. We show that our methods are applicable to various NLMs (BERT, XLNet, RoBERTa, and ALBERT), as well as alternative ontologies (UMLS from the medical domain, and multilingual WordNet). Our contributions also include an in-depth comparison of WSD approaches, showing that

k-NN, with sense embeddings and self-supervised NLMs, outperforms fine-tuned NLMs in few-shot settings while exhibiting less bias towards the most frequent senses.

Our findings support that internal states from NLMs can be reliably employed for symbolic representation of concepts featured in unstructured texts. While this thesis is limited to exploring applications of these representations using *k*-NN, it stands to reason that more sophisticated methods, such as PL, may also benefit from them. We hope this thesis helps prepare some of the groundwork for the development of neurosymbolic hybrid approaches driving the next set of breakthroughs in NLP.

Resumo

Os últimos desenvolvimentos em Modelos Neuronais de Linguagem (MNLs) de grandes dimensões, baseados em aprendizagem computacional profunda *end-to-end*, foram responsáveis por avanços sem precedentes na área de Processamento de Linguagem Natural (PLN). Notavelmente, este progresso tem sido impulsionado por uma *pipeline* básica que consiste em inicialmente treinar um MNL, sem supervisão, em grandes quantidades de textos (i.e., pré-treino), seguido pelo ajuste do mesmo MNL para variadas tarefas usando conjuntos de dados específicos a essa tarefas, de dimensão relativamente reduzida. Consequentemente, repetindo este processo recorrendo a MNLs com maior número de parâmetros obtemos melhorias nas suas prestações transversais a várias tarefas.

No entanto, espera-se que este melhoramento devido à escalada da quantidade de dados e recursos computacionais venha a estagnar no futuro próximo, e soluções hibridas que combinam MNLs com abordagens complementares, como a Lógica Probabilistica (LP), têm sido ativamente investigadas. Nesta configuração híbrida, assume-se que MNLs sejam responsáveis pela produção de representações contextuais que possam ser utilizadas por modelos complementares para raciocínio lógico, uma conhecida limitação dos MNLs. A natureza destas representações, bem como a interface entre MNLs e PL, tratam-se de importantes questões em aberto na investigação de PLN.

Esta tese explora formas de extrair representações ao nível de sentidos lexicais a partir dos estados internos destes MNLs, tendo em visto o desenvolvimento de soluções híbridas que possam beneficiar de MNLs para representação simbólica de conhecimento de senso comum. Os métodos que propomos, baseados em *embeddings* contextuais, permitem associar palavras e expressões inseridas em textos não estruturados ao conjunto de conceitos relativos a senso comum contidos numa popular ontologia, a WordNet. Em vez de realizar ajustes a estes modelos específicos a determinadas tarefas, ou treinar outros modelos baseados em atributos dos MNLs, a nossa abordagem explora o espaço latente resultante do pré-treino auto-supervisionado de MNLs, recorrendo ao método de vizinhos mais próximos (k-VP) para inferência. Adicionalmente, exploramos formas de utilizar as nossas representações para melhorar extração de relações *zero-shot* a partir de MNLs, com foco em relações relevantes para raciocínio de senso comum (e.g., *UsadoPara*).

Avaliamos o nosso progresso de acordo com a prestação dos nossos métodos em várias tarefas relacionadas com Desambiguação Lexical de Sentido (DLS), reportando resultados

estado-de-arte em algumas destas tarefas. Mostramos que os nossos métodos aplicam-se a vários MNLs (BERT, XLNet, RoBERTa, e ALBERT), bem como ontologias alternativas (UMLS relativa ao domínio médico, e WordNet multilingue).

Os nossos contributos também incluem uma comparação aprofundada entre diferentes abordagens de DLS, revelando que *k*-VP, usando *embeddings* de sentido e MLNs auto-supervisionados, consegue melhores resultados do que ajustes de MLNs em cenários *few-shot*, demonstrando também menor enviesamento para os sentidos mais frequentes.

As nossas descobertas reforçam que os estados internos de MNLs podem ser aplicados fiavelmente para representação simbólica de conceitos presentes em textos não estruturados. Embora esta tese limite-se à exploração de aplicações destas representações usando *k*-VP, achamos expectável que métodos mais sofisticados, como LP, possam também beneficiar da sua aplicação. Esperamos que esta tese ajude a preparar a fundação necessária para o desenvolvimento de abordagens híbridas neurosimbólicas que impulsionem o próximo conjunto de avanços em PLN.

Contents

Glossary

CSK	Commonsense Knowledge
FOL	First-order Logic
GWSC	Graded Word Similarity in Context
LMMS	Language Modelling Makes Sense
MRR	Mean Reciprocal Rank
NLI	Natural Language Inference
NLM	Neural Language Model
NLP	Natural Language Processing
PL	Probabilistic Logic
SID	Sense Identification Dataset
SP	Sense Profile
SRL	Semantic Role Labeling
USM	Uninformed Sense Matching
UWA	Unambiguous Word Annotations
WSD	Word Sense Disambiguation
WiC	Word-in-Context
k-NN	*k*-Nearest Neighbors

Chapter 1

Introduction

For the past decade, neural approaches (i.e., deep learning) have driven most progress on Artificial Intelligence (AI), powering various achievements from AlexNet [1] to GPT-3 [2]. The gradient-based approach of deep learning, along with increasing computational resources, has proven extraordinarily effective at learning useful models from raw data, without need for handcrafted features. Still, the current paradigm of end-to-end deep learning raises concerns regarding their poor explainability and modularity, causing many to question their reliability [3–5]. It is also not clear how far the current gains obtained by training larger models on more data can extend into the future [6–8]. In contrast, a symbolic approach, particularly formal logic, is naturally suited for logical reasoning, a known limitation of Neural Language Models (NLMs) [9], and is designed for provable correctness*. However, standard inference mechanisms of logic-based approaches are not designed for learning symbolic representations from raw data, making this approach too brittle for the richness of natural language, and real-world applications broadly. Moreover, symbolic approaches have historically faced challenges with intractable search spaces [10, 11].

Neurosymbolic AI has emerged with the goal of combining the best of neural and symbolic approaches. Neurosymbolic approaches can take various shapes, combining neural and symbolic methods in different manners, and at varying degrees of integration [12, 13]. From these various possibilities, our work in this thesis is most relevant for shallow hybrids, with clear separation between neural and symbolic components. This particular variety of hybrid approach, aligned with recent findings in neuroscience [14], is interested in using neural methods for representation learning, and symbolic methods for

*The causal chain of reasoning steps behind each inference can be trivially inspected.

reasoning and explainability. Our contributions towards this hybrid are strictly focused on the development of the neural component, following prior work in assuming that existing logic-based inference methods may suffice for the reasoning required in some Natural Language Processing (NLP) tasks [15]. Consequently, research into logic-based inference and related applications is considered beyond the scope of this thesis, and we focus on neural-based approaches for learning natural language representations which may be employed by such hybrids in future work (see Chapter 3). More specifically, we focus on sense-level distributional representations based on the latest NLMs, considering that sense-level representations are particularly relevant for commonsense reasoning – a major target for research into hybrid approaches [16, 17].

Related work on distributional sense representations, or sense embeddings, has focused on providing a solution to the so-called Meaning Conflation Deficiency [18] of traditional word embeddings, which merge different meanings into the same word-level representation. Most works have explored variations on the popular word2vec [19] method for producing sense-level embeddings [20–23], but the dynamic word-level interactions composing sentential context were not targeted by those works, which has been shown to be crucial for meaning understanding in humans [24]. It would take the development of large NLMs (particularly BERT [25]), and corresponding contextual embeddings, until state-of-the-art sense embeddings could become accurate enough to rival supervised systems for Word Sense Disambiguation (WSD), for example.

In this thesis we explore how to best leverage the latest NLMs for state-of-the-art sense embeddings. We also evaluate our sense embeddings on classical NLP tasks, like WSD, and additional sense-related tasks that allow us to measure progress towards accurate and versatile sense representations. Conversely, we also explore how research into sense embeddings can provide insights on intrinsic properties of NLMs, such as layer specialization, or acquisition of commonsense knowledge from pre-training.

Organization Having already described the motivation behind our work in the above, we introduce our research questions (Section 1.1) and list of publications (Section 1.2) in the next sections. The remainder of this cumulative thesis provides a focused description of our contributions in aggregate (combining findings from different publications) in Chapter 2, followed by a reflective appreciation of our efforts (Chapter 3), and our final conclusions (Chapter 4). We include our publications related to this thesis as appendices (see Table of Contents).

1.1 Research Questions

In this thesis our main concern is to assess the extent to which recent large NLMs can be used for sense representation and matching, towards a neurosymbolic hybrid solution that uses NLMs for concept representation and formal logic for reasoning and inference. As such, we formulate our central research question as the following:

Can we use Transformer-based large NLMs to accurately map free-form text spans to precisely-defined commonsense concepts?

This pursuit raises a more specific set of related research questions, namely:

RQ1 How can we leverage pre-trained NLMs for canonical sense representation and matching?

RQ2 Which tasks directly benefit from improved sense representations, and how can they be used to measure progress in this direction?

RQ3 Can we explain which aspects of NLMs most contribute towards accurate sense representation from self-supervised learning?

RQ4 Which additional tasks can benefit from our improved sense representations?

1.2 List of Publications

In this section, we list the publications produced in the scope of this thesis, and relate them to the research questions described earlier. Each publication listed below is included in this thesis at the corresponding appendix. Individual contributions from these publications are covered in more depth in Chapter 2. We group our publications into three categories, as illustrated in Figure 1.1:

1. **Sense Representation:** Our main publications proposing methods for improved sense representations, and evaluating them on the tasks most relevant for our goals [26–28] – RQ1-3.

2. **Word Sense Disambiguation:** Complementary publications exploring applications of sense embeddings for WSD in more depth [29–31] – RQ2.

3. **Use Cases:** Publications exploring use cases which are not directly related to sense representation, but still benefit from our work [32, 33] – RQ4.

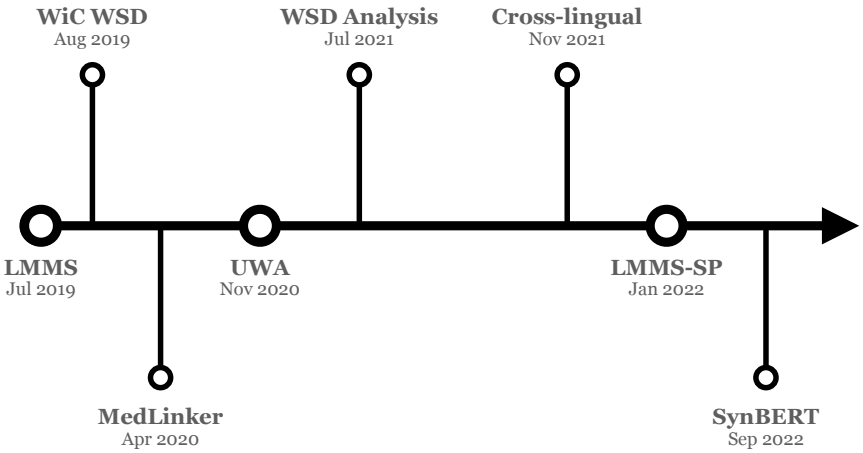


FIGURE 1.1: Timeline illustrating how our publications (using labels from below) relate to our main theme of sense representation, positioned along a central axis. References above the axis are more related to WSD, while those below address additional use cases.

LMMS [26, RQ1-2] Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation. ACL 2019. Appendix A.

WiC WSD [29, RQ2] LIAAD at SemDeep-5 Challenge: Word-in-Context (WiC). SemDeep-5. Appendix B.

MedLinker [32, RQ4] MedLinker: Medical Entity Linking with Neural Representations and Dictionary Matching. ECIR 2020. Appendix C.

UWA [27, RQ1] Don't Neglect the Obvious: On the Role of Unambiguous Words in Word Sense Disambiguation. EMNLP 2020. Appendix D.

WSD Analysis [30, RQ3] Analysis and Evaluation of Language Models for Word Sense Disambiguation. Computational Linguistics 2021. Appendix E.

Cross-lingual [31, RQ1] On the Cross-lingual Transferability of Contextualized Sense Embeddings. MRL 2021. Appendix F.

LMMS-SP [28, RQ1,3] LMMS Reloaded: Transformer-based Sense Embeddings for Disambiguation and Beyond. Artificial Intelligence Journal 2022. Appendix G.

SynBERT [33, RQ4] Precisely Probing Commonsense Knowledge in Pretrained Language Models using Sense Embeddings. Under Review. Appendix H.

Chapter 2

Contributions

We begin this chapter by providing background information on the main topics addressed in this thesis (Section 2.1), from the foundations of vector-based meaning representations to the various sense-related tasks we cover.

Afterwards, we explain our methodology for sense representation with Neural Language Models (NLMs) in Section 2.2, addressing RQ1, based on our work in Loureiro and Jorge [26], Loureiro and Camacho-Collados [27], and Loureiro et al. [28].

In Section 2.3, we describe how we apply sense embeddings for various sense-related tasks (RQ2), as well as relation extraction (RQ4) and other potential applications which did not get the opportunity to be explored in the scope of this thesis. Our methods for applying sense embeddings are generally covered in Loureiro et al. [28], and more specifically Loureiro and Jorge [29], Loureiro and Jorge [32] and Loureiro and Jorge [33].

Still related to applications of our sense embeddings, in Section 2.4 we highlight the layer-wise probing experiments of Loureiro et al. [28] (RQ3), along with our work on probing CSK learned from pre-training (RQ4), as explored in Loureiro and Jorge [33].

Finally, in Section 2.5 we report our results on the various sense-related tasks covered in our work, and related works proposing alternative sense embeddings.

The open-source code, datasets, and sense embeddings relative to the contributions described in this thesis are available at `https://github.com/danlou/LMMS`.

2.1 Background

Vector Semantics

As far back as 1935, Firth [34] postulated that "the meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously". Indeed, after working on formal theories of word meaning definition, Wittgenstein [35] conceded "the meaning of a word is its use in a language". This view of meaning representation became known as the Distributional Hypothesis [36], which proposes that words that occur in the same contexts tend to have similar meanings. Until the turn of the century, distributional representations of words were essentially based on word-document weighted frequency matrices, developed for Information Retrieval applications [37–42]. Nevertheless, Schutze [43] and Yarowsky [44] had already realized the potential for Word Sense Disambiguation (WSD) applications based on the similarity between unsupervised word embeddings.

A milestone in the evolution of word embeddings was the discovery that NLMs implicitly develop word embeddings when training for the task of word prediction [45]. Shortly after this discovery, [46–48] demonstrated that word embeddings could be incorporated into neural architectures designed for various NLP tasks. With word2vec, Mikolov et al. [49] distilled the components of NLMs responsible for learning word embeddings into a lightweight and scalable solution, allowing this neural-based solution to be employed on corpora of unprecedented size (100B tokens). Nevertheless, count-based solutions remained popular, particularly GloVe [50]. The next major improvement was fastText [51], which could represent words absent from training data using subword information, besides refining word2vec's training method.

In spite of their success, word2vec, GloVe and fastText conflated different senses of the same word form into the same representation, a shortcoming known as the Meaning Conflation Deficiency [18]. While a number of extensions were proposed for the creation of sense-specific representations, such as AutoExtend [20], NASARI [52], DeConf [23] or Probabilistic FastText [53], this issue would require the development of a new generation of NLMs in order to be effectively addressed.

Neural Language Modelling

The first major step towards contextual embeddings from NLMs, was the development of context2vec [54], a single-layer bidirectional LSTM trained with the objective of maximizing similarity between hidden states and target word embeddings, similarly to word2vec. Peters et al. [55] built upon context2vec with ELMo, a deeper bidirectional LSTM trained with language modelling objectives that produce more transferrable representations. Both context2vec and ELMo emphasized WSD applications, providing the most convincing accounts until then that sense embeddings can be effectively represented as centroids of contextual embeddings, showing 1-NN solutions to WSD tasks that rivalled the performance of task-specific models.

With the introduction of highly-scalable Transformer architectures [56], two kinds of very deep self-supervised NLMs emerged: causal (or left-to-right) models, epitomized by GPT-3 [2], where the objective is to predict the next word given a past sequence of words; and masked models, where the objective is to predict a masked (i.e., hidden) word given its surrounding words, of which the most prominent example is BERT [25]. The difference in training objectives results in these two varieties of NLMs specializing at different tasks, with causal models excelling at language generation and masked models at language understanding. BERT proved highly successful at most NLP tasks [57] and motivated the development of numerous derivative models. Below we provide details about each of the NLMs we experimented with, highlighting their differences.

BERT The model released by Devlin et al. [25] is first prominent Transformer-based NLM designed for language understanding. It is pre-trained with two self-supervised modelling objectives, Masked Language Modelling (MLM) and Next Sentence Prediction (NSP), using English Wikipedia and BookCorpus [58]. It uses WordPiece tokenization, splitting words into different components at the character-level (i.e., subwords). BERT is available in several models differing not only on parameter size, but also tokenization and casing.

XLNet Based on a Transfomer-XL [59] architecture, Yang et al. [60] release XLNet featuring Permutation Language Modelling (PLM) as the only pre-training objective. The motivation for PLM is that it does not rely on masked tokens, and thus makes pre-training closer to fine-tuning for downstream tasks. It is also trained on much larger corpora than

BERT, adding a large volume of web text from various sources to the corpora used for BERT. Instead of using WordPiece for tokenization, XLNet uses SentencePiece [61], which is a very similar open-source version of WordPiece.

RoBERTa The model proposed by Liu et al. [62] is explicitly designed as an optimized version of BERT. RoBERTa does not use the NSP pre-training objective after finding that it deteriorates performance in the reported experimental setting, performing only MLM during pre-training. It is also trained with some different choices of hyperparameters (e.g., larger batch sizes) that improve performance on downstream tasks. The models released with RoBERTa are also trained on larger corpora composed mostly of web text, similarly to XLNet. As for tokenization, RoBERTa opts for byte-level BPE, following Radford et al. [63], which makes retrieving embeddings for specific tokens more challenging (i.e., spacing must be explicitly encoded).

ALBERT Aiming for a lighter architecture, Lan et al. [64] propose ALBERT as a more parameter-efficient version of BERT. In spite of changes introduced to improve efficiency (e.g., cross-layer parameter sharing), ALBERT is based on a similar architecture to BERT. Besides improving efficiency, ALBERT also improves performance on downstream tasks by replacing NSP with the more challenging Sentence Order Prediction (SOP) objective. ALBERT uses the same SentencePiece tokenization as XLNet, and it is trained on similar corpora. It is released in several configurations, showing benchmark performance comparable to BERT while using fewer parameters.

Sense Inventory

The currently most popular English word sense inventory is the Princeton WordNet* [65] (henceforth, WordNet), a large semantic network comprised of general domain concepts curated by experts. The core unit of WordNet is the synset, which represents a cognitive concept. Each lemma (word or multi-word expression) in WordNet belongs to one or more synsets, and word senses amount to the combination of word forms and synsets (referred as sensekeys). The predominant semantic relation in WordNet, which relates synset pairs, is hypernymy (i.e., Is-A). Each synset also features a gloss (dictionary definition), part-of-speech (noun, verb, adjective or adverb) and supersense, which is a syntactic category and logical grouping.

For example, the lemma 'mouse' is polysemous belonging to the $mouse_n^1$ (rodent) and $mouse_n^4$ (computer mouse) synsets, among others. Its most frequent sense, mouse%1:05:00:: (sensekey), belongs to the synset $mouse_n^1$ which has an hypernymy relation with $rodent_n^1$, supersense 'noun.animal', and gloss "any of numerous small rodents typically [...]".

Commonsense Knowledge

Commonsense Knowledge (CSK) consists of knowledge about the everyday world that is universally accepted, considered obvious, and thus, not usually explicitly stated [66, 67]. The most prominent CSK resource is ConceptNet [68], which expresses CSK as triples where text fragments represent concepts, and these concepts are interconnected through 20 relation types (e.g. "sleeping" *Causes* "being refreshed").

ConceptNet was developed by an extensive crowdsourcing effort and it's been widely adopted for various NLP tasks. As commonsense reasoning becomes an increasingly popular topic for state-of-the-art research, ConceptNet has played a central role both in developing systems that can leverage its semantic network, and in creating new challenging tasks designed to target this sort of reasoning abilities specifically (e.g., [16, 69]). ConceptNet is also one of the main resources used for the LAMA [70] probing task, which evaluates the CSK learned by NLMs from self-supervised pre-training.

As a consequence of relying on free-form text for representing its nodes, rather than disambiguated (canonical) representations (e.g., synsets), ConceptNet allows for redundant and misleading associations which limit generalization [71], besides aggravating the network's sparsity [72, 73].

*We use WordNet v3.0, with 117,659 synsets, 206,949 senses, 147,306 lemmas, and 45 supersenses.

Sense-related Tasks

In this thesis, we address several sense-related tasks selected to investigate the versatility of the proposed sense embeddings, covering disambiguation (WSD), matching (USM), meaning change detection (WiC, GWSC, and SCWS), and sense similarity (SID). Tasks related to disambiguation are addressed using SP-WSD (WSD, WiC, GWSC and SCWS), while tasks more related to matching or comparing without lexical constraints are addressed using SP-USM (USM and SID), according to the findings in Loureiro et al. [28].

All tasks are solved using cosine similarity between contextual embeddings and LMMS-SP precomputed sense embeddings represented using the same NLM. No additional task-specific training or validation datasets are used asides from those referred in Section 2.2, and all NLMs are employed in the same fashion – simply retrieving contextualized representations from each layer. In Loureiro et al. [28], we provide more details about how these similarities are used to produce task-specific predictions – essentially minor variations on the methods presented in Section 2.3. As such, performance on these tasks should be indicative of each NLM's intrinsic ability to approximate meaning representations learned during pre-training with language modelling objectives alone.

Below we provide a short introduction to each of these tasks.

WSD Word Sense Disambiguation (WSD) is a classical NLP task considered to be AI-complete [74]. It consists in assigning the correct sense to an ambiguous word in a given context, out of predefined inventory of word senses. In additional to sentential context, WSD tasks usually also provide the word's lemma and part-of-speech (POS). The compilation of test sets included in Raganato et al. [75] as become of the de facto evaluation framework for English WSD. Performance on WSD is measured with the F1 metric.

USM The Uninformed Sense Matching (USM) task introduced in Loureiro and Jorge [26] is a variation on WSD that can more accurately represent the extent to which NLMs can associate words or phrases to senses from the WordNet inventory. The crucial difference in relation to WSD is that USM does not use any supplemental information to restrict candidates in the sense inventory. This conveniently allows USM to use the same test sets as WSD. Performance on USM is measured with ranking metrics, namely Precision at 5 (P@5) and Mean Reciprocal Rank (MRR).

WiC The Word-in-Context [76, WiC] task is designed to assess how context impacts word representations produced by contextual NLMs. It is a binary classification task that simply requires determining whether a particular word is used with the same meaning or not in a pair of sentences, also given lemma and POS provided in WSD tasks. The dataset is balanced and performance is measured with accuracy.

GWSC Unlike the binary contextual similarity assignments of WiC, with Graded Word Similarity in Context [77, GWSC] we're evaluating graded contextual similarity. GWSC targets word pairs used for evaluating distributional semantic models (not necessarily polysemous words) in contexts spanning multiple sentences. The task is divided into two sub-tasks derived from human-annotated similarity ratings: 1) predict the change in similarity between two different contexts for each word pair; 2) predict the similarity ratings themselves. On this thesis we focus on sub-task 2, which is measured with the harmonic mean of the Spearman and Pearson correlations between the system's scores and the average human annotations.

SCWS The Stanford Contextual Word Similarities [78, SCWS] task is the inspiration for GWCS. With SCWS, we are provided two words in context, each within an independent sentence, and need to predict their graded contextual similarity. Performance is measured with Spearman correlation between predicted similarity and human ratings.

SID All previously discussed tasks in this thesis evaluate sense embeddings by their utility for accurately matching or distinguishing word senses in particular contexts. In this last task, we address intrinsic evaluation of sense embeddings, directly comparing cosine similarity between sense pairs against human similarity ratings. We perform this evaluation using the Sense Identification Dataset [79, SID], which is based on word pairs (nouns only) and human similarity ratings from SemEval-2017 Task 2 [80], with the addition of mapping word pairs to particular senses in the BabelNet sense inventory. We map word senses to WordNet and measure performance using Pearson correlation.

2.2 From NLMs to Sense Representation

Seed set of Representations

The first step in our process to learn sense embeddings using NLMs is based on contextual embeddings corresponding to sense-annotated corpora (see Figure 2.1).

Essentially, in order to generate sense embeddings learned in context from natural language, we require a pre-trained contextual NLM Ω (frozen parameters) and a corpus of sense-annotated sentences S. Every sense ψ is represented from the set of contextual embeddings $\vec{c}_l \in C_\psi$, obtained by employing Ω on the set of sentences S_ψ annotated with that sense (considering only contextual embeddings specific to tokens annotated with sense ψ), using representations at each layer $l \in L$, such that:

$$\vec{\psi} = \frac{1}{|C_\psi|} \sum_{l \in L} \sum_{\vec{c} \in C_\psi} \vec{c}_l \text{ , where } C_\psi = \Omega(S_\psi) \tag{2.1}$$

The set of layers L used with LMMS [26] was the last four $[-1, -2, -3, -4]$ (reversed layer indices), following BERT's original paper [25] which reports best results using this specific set of layers for Named Entity Recognition. The same pooling (i.e., sum of the last four layers) has since become the default for other WSD works [81–84].

However, with LMMS-SP [28], we aim for a more principled approach that better supports the choice of layers used for pooling contextual embeddings and does so in a generalizable manner that can find another optimal set of layers specific to any given NLM. This principled approach involves an analysis which probes each layer's adeptness for sense representation, followed by a method for setting layer specifc weights for pooling. We distinguish between weights for sense disambiguation and matching following findings in Loureiro et al. [28] regarding the differences in these two representation modes. Consequently, these weights are specific to NLMs and their intended applications, and we denote them Sense Profiles (SP) - SP-WSD for disambiguation and SP-USM for matching.

In Loureiro and Camacho-Collados [27], we introduce the UWA corpus, extending sense-annotated WordNet coverage from 16% to 53%, when combined with SemCor [85] the largest WordNet sense-annotated corpus available. UWA targets exclusively sense-annotations for unambiguous words, extracted automatically from large corpora. This increase in coverage reduces the number of high-density clusters resulting from coverage extension through propagation methods, and improves representations of senses with ambiguous words from network effects.

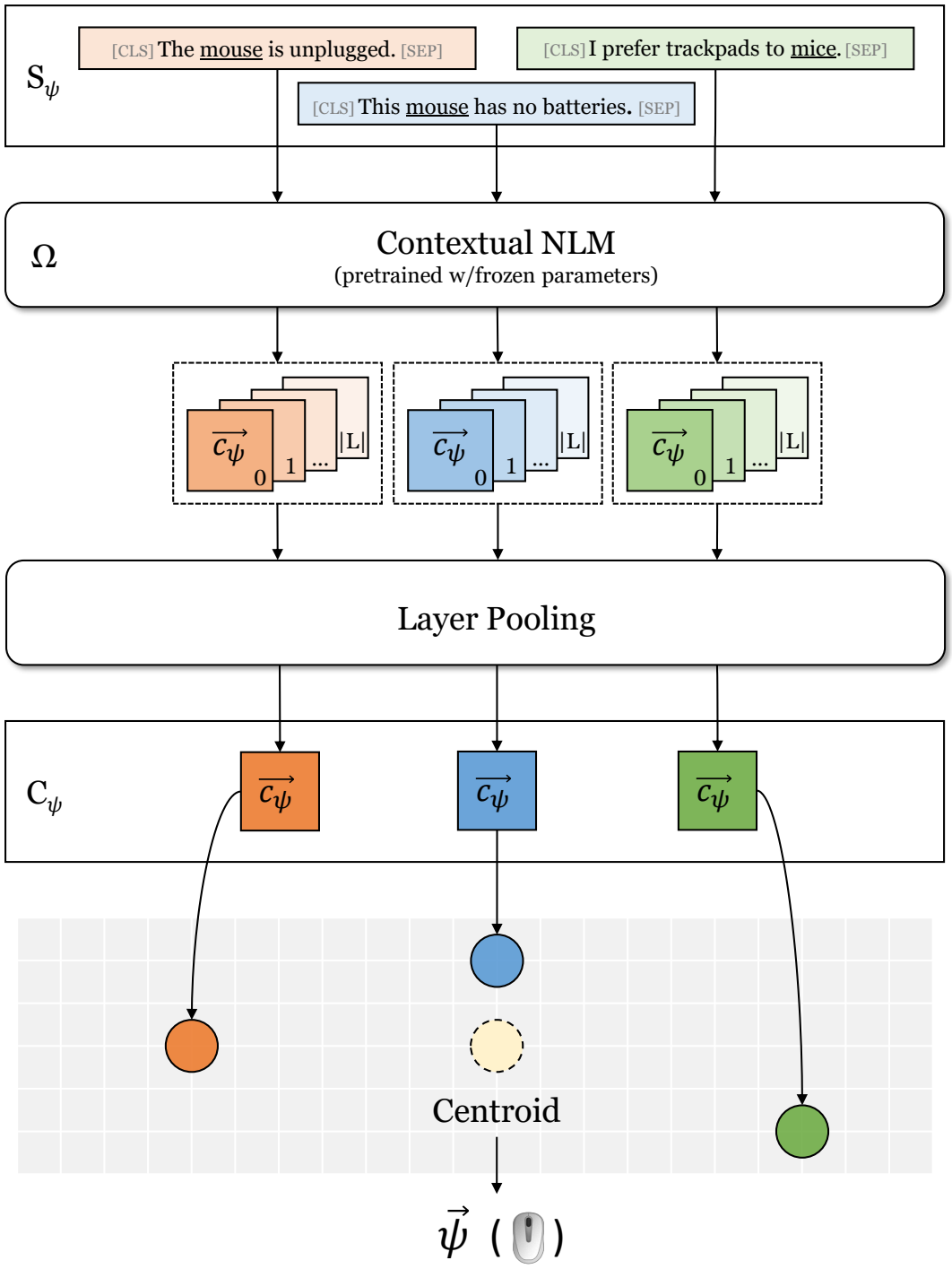


FIGURE 2.1: Overview of learning sense embeddings from annotated corpora. Showing how the sense ψ for 'computer mouse' is determined from a set for sentences annotated with that sense S_ψ (padded with special tokens). After pooling contextual embeddings C_ψ from layers L, the sense embedding for $\vec{\psi}$ is computed as the centroid of C_ψ.

Propagating to Full-Coverage

Since available corpora do not provide full-coverage annotations for every sense in WordNet, we require an alternative procedure to represent the remaining set of senses.

In Loureiro and Jorge [26], we show that it is possible to infer remaining sense embeddings without annotations, from an initial subset of sense embeddings, along with relations present in WordNet. Our proposed propagation process involves three steps, using increasingly abstract relations from WordNet - sets of synonyms (synsets), hypernymy relations, and lexical categories (supersenses). Unrepresented senses are inferred at each sequential step as the average of sense embeddings that share the relation corresponding to that step. This method ensures full-coverage provided that initial sense embeddings are sufficiently diverse such that falling back on propagating from supersenses is always possible.

However, this approach is susceptible to the creation of high-density clusters in the embedding space when several senses are represented from the same set of previously represented senses, effectively resulting in a coarser set of sense embeddings that's unhelpful for disambiguation or matching applications. This unintended clustering is mitigated using our proposed UWA [27] corpus (see Figure 2.2), explained earlier, and by leveraging glosses and lemmas in alternative to annotations as explained next.

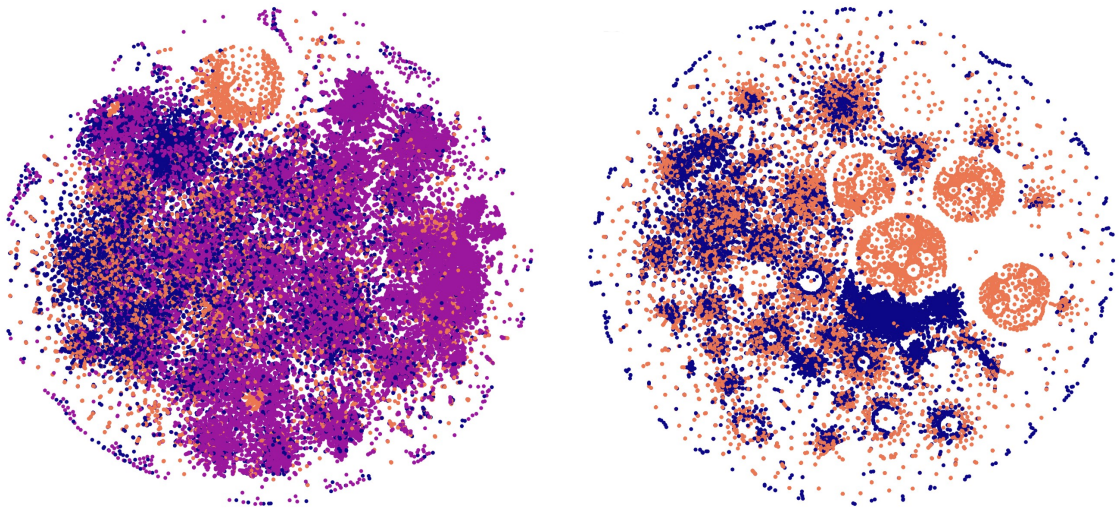


FIGURE 2.2: T-SNE [86] comparison of synset embeddings for whole WordNet learned from SemCor (SC) augmented with UWA (left), or just SC (right). Colors represent source of annotations for embeddings (● SC ● UWA ● Propagation). Illustrates the extent to which UWA helps to reduce dense clustering from propagation in the embedding space.

Glosses and Lemmas

In Loureiro and Jorge [26], we introduce a method for representing sense embeddings based on glosses and lemmas, independently from sense-annotated corpora.

This method is inspired by a typical baseline approach used in works pertaining to sentence embeddings, and it amounts to simply averaging the contextual embeddings for all tokens present in a sentence. In our case, we use glosses as sentences, but also introduce lemmas into the gloss' context (i.e., "<*lemma*> , <*sense lemmas*> - <*gloss*>"). By combining glosses with lemmas, we not only augment the information available to represent senses, but we are also able to generate sense embeddings which are lemma-specific (sensekey-level), instead of only concept-specific (synset-level) if we only used glosses. As such, sense embeddings generated by this method address the redundancy issue arising from the previously described propagation method, while simultaneously introducing representational information which is complementary to contextual embeddings extracted from sense-annotated sentences.

While Loureiro and Jorge [26] proposes using concatenation to merge this new set of sense embeddings based on glosses and lemmas with the previously mentioned set, in Loureiro et al. [28] we propose merging through averaging instead. This departure is motivated by the fact that in spite of Loureiro and Jorge [26] reporting that concatenation outperforms averaging for WSD, the difference in performance was modest, and an extensive analysis on Loureiro et al. [28, pp. 52] finds that averaging produces better results in additional tasks. Interestingly, this analysis also shows that gloss embeddings can be competitive on some tasks when compared to sense embeddings learned from annotations and propagation. Merging representations through concatenation doubles the dimensionality of sense embeddings, increasing computational requirements and adding complexity to potential applications. On the other hand, merging representations through averaging allows for adding more components while retaining a similar vector, of equal dimensionality to contextual embeddings, and represented in the same vector space.

Furthermore, in Loureiro and Jorge [33], we show that template-based relation extraction [33] with glosses mentioned explicitly within templates produces substantially more accurate predictions (i.e., "<*synset*> can be defined as : <*gloss*> . `[SEP]` <*assertion*>").

2.3 Applications of Sense Embeddings

Sense Disambiguation or Matching

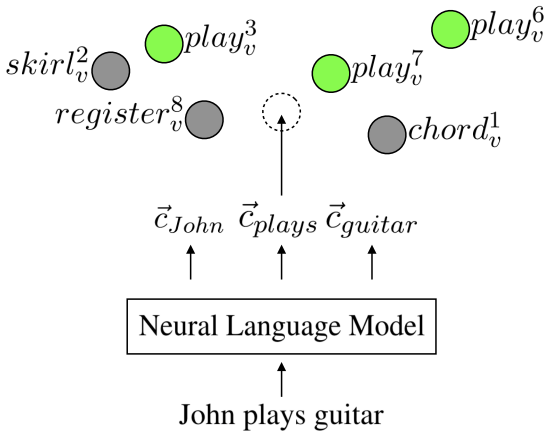


FIGURE 2.3: Illustration of our *k*-NN approach in which sense embeddings (pre-computed) are represented in the same space as contextual embeddings. Green nodes belong to the same subset of lemma and part-of-speech (relevant for disambiguation). Grey nodes correspond to different subsets, which can be closer (relevant for matching).

We consider two types of sense embeddings according to their intended types of application: disambiguation or matching. Disambiguation assigns a word in context (i.e., in a sentence) to a particular sense out of a subset of candidate senses, restricted by the word's lemma and part-of-speech. Matching also assigns specific senses to words, but imposes no restrictions, admitting every entry in the sense inventory for each assignment.

The different conditions for disambiguation and matching require sense representations with different degrees of lexical information and semantic coherence. Whereas, for disambiguation, lexical information can be absent from sense representations, due to the subset restrictions, for matching, lexical information is essential to distinguish between word forms carrying identical or similar semantics. Similarly, the disambiguation setting has no issues with sense representations displaying inconsistencies such as *eat* being more similar to *sleep* than to *drink*, since these all belong to disjoint subsets, but the order and coherence of these similarities is relevant for sense matching applications. Thus, both disambiguation and matching is based on cosine similarities between a word's contextual embedding and pre-computed sense embeddings. Provided both embeddings types are computed using the same layer pooling (according to application), the difference between disambiguation and matching amounts to whether we restrict sense embeddings to a subset specific to particular lemma and part-of-speech or not (see Figure 2.3).

Combining Contextual and Sense Similarity

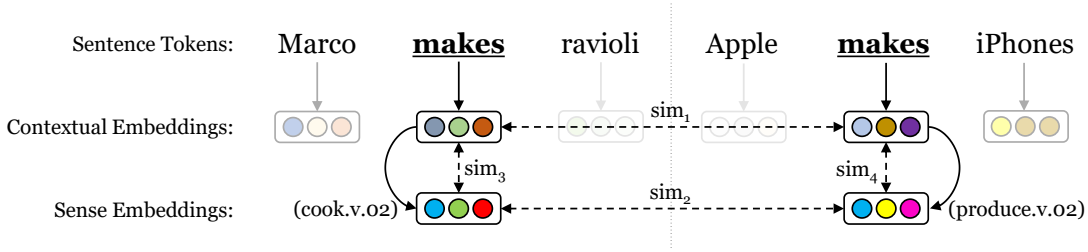


FIGURE 2.4: Components and interactions involved in our approaches. The sim_n labels correspond to cosine similarities between the related embeddings. Sense embeddings obtained from 1-NN matches of contextual embeddings.

For some tasks, the goal is to detect whether a word occurring in two distinct contexts is referring to the same sense, without necessarily needing to predict particular senses from a pre-defined inventory. This is precisely the case for the WiC [76] binary classification task that we address in our work, along with GWSC [77] and SCWS [78] which are slight variations requiring graded similarity scores instead (more details in Section 2.1).

Generically, given contexts A and B, we disambiguate target words in the corresponding contexts using the 1-NN approach described earlier, and compute sense similarities sim^A_{wsd} and sim^B_{wsd} as the cosine similarity between the embeddings of the predicted senses. Considering that disambiguation may predict the same senses, thus potentially resulting in $sim^A_{wsd} = sim^B_{wsd}$ for many instances, we also compute contextual similarities sim^A_{ctx} and sim^B_{ctx} as the cosine similarity between the contextual embeddings of the target words. Thus, we determine similarity scores specific to context A as $sim^A = \frac{1}{2}(sim^A_{wsd} + sim^A_{ctx})$, and similarity scores specific to context B as $sim^B = \frac{1}{2}(sim^B_{wsd} + sim^B_{ctx})$. As such, graded similarity changes are simply computed as $sim^B - sim^A$.

In Loureiro and Jorge [87] we experiment with additional sense and contextual similarity combinations (see Figure 2.4), and propose a supervised approach for the WiC task using these similarities as the only features for a Logistic Regression binary classifier. Nevertheless, our default approach for the WiC task, as reported in Loureiro et al. [28], is unsupervised and based exclusively on the outcome from the disambiguation step.

While not as relevant for sense representation in specific, in Loureiro and Jorge [32] we also explore combining embedding similarity with scores produced by Approximate Dictionary Matching [88] methods. This work on entity linking also demonstrates that our approach is applicable to alternative ontologies, particularly UMLS.

Enhancing NLMs with Sense Embeddings

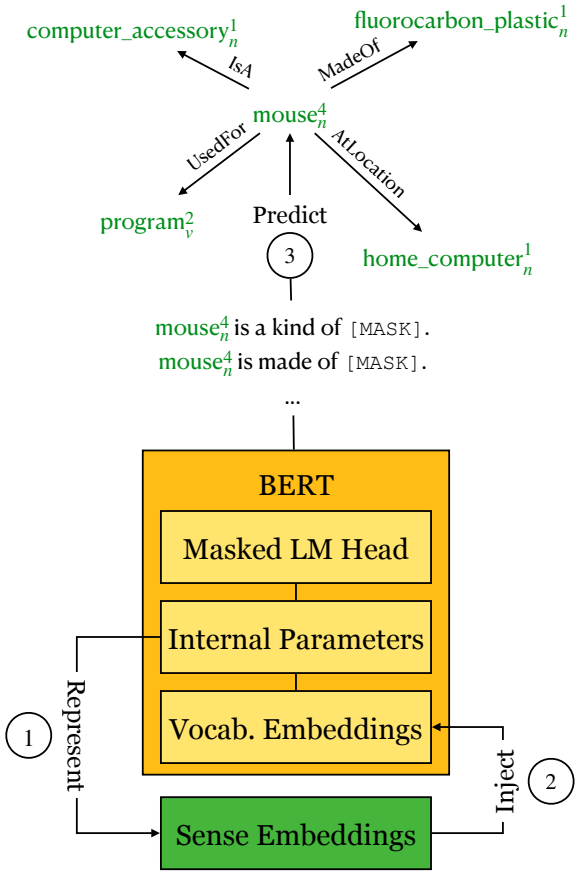


FIGURE 2.5: Our 3-step method for extracting unsupervised commonsense relations between concepts (i.e., word senses) from pre-trained NLMs. Relations are expressed as verbalizations that may be exchanged to target any other property of interest.

In Loureiro and Jorge [33] we propose enhancing NLMs with sense embeddings learned from their own internal states. The integration of explicit sense-level representations at the vocabulary-level of NLMs enables their use for various tasks as if they were regular tokens from the NLM's vocabulary, including for masked predictions (see Figure 2.5).

The most direct application of these NLMs enhanced with sense embeddings is for probing commonsense knowledge learned during pre-training with higher precision and no vocabulary restrictions, a topic that has gathered much research interest lately [70, 89, 90]. In Loureiro and Jorge [33], we also propose the SenseLAMA dataset specifically designed for probing the set of commonsense relations featured in ConceptNet [68], but using WordNet synsets as arguments. Another application explored in Loureiro and Jorge [33] is zero-shot commonsense relation extraction. Based on the same infilling cloze-style approach used for probing CSK, we can discover new triples grounded in WordNet for any relation that can be verbalized as a short assertion (see Predict step of Figure 2.5).

Additional Potential Applications

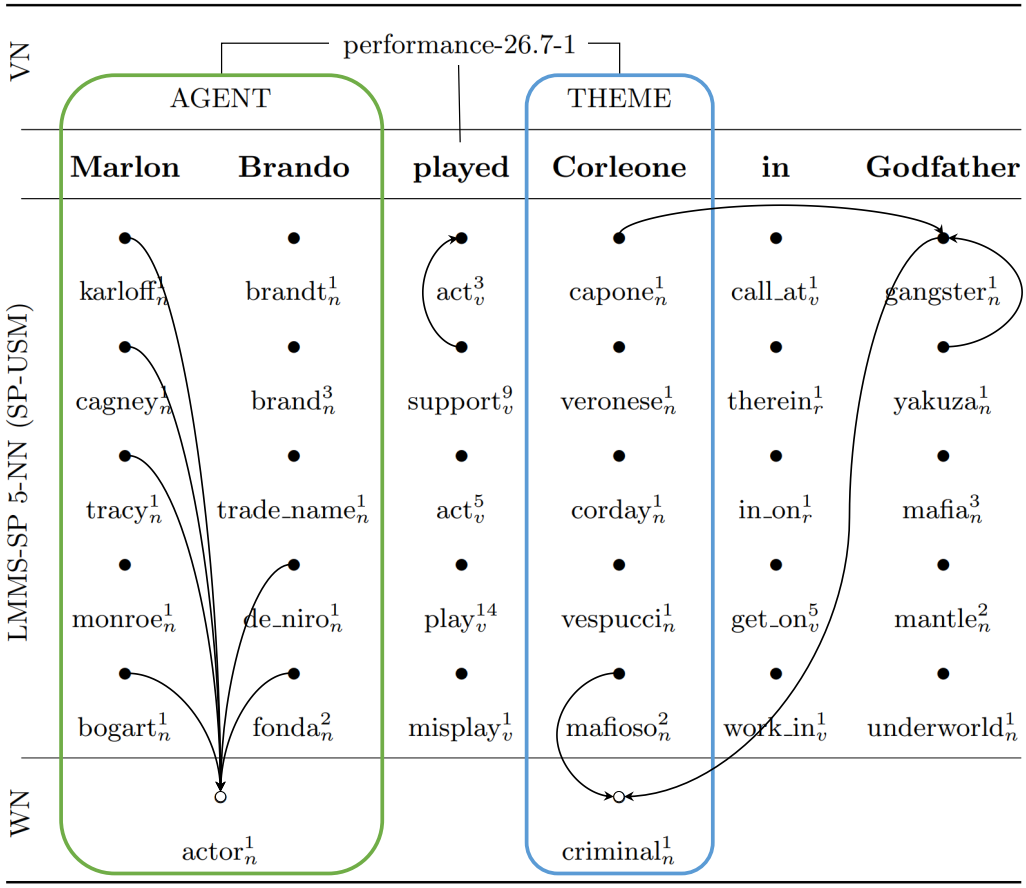


FIGURE 2.6: Example sentence with each token matched to LMMS-SP$_{\text{ALBERT-XXL}}$ sense embeddings, presenting synsets for the 5 nearest neighbors. Shows direct hypernymy relations (i.e., Is-A), included in WordNet (WN), between matched synsets, along with relations shared between more than one matched and unmatched synset (i.e., deducible generalizations). Finally, at the top, we show a VerbNet (VN) semantic frame matched to this sentence, highlighting how LMMS-SP enables generalization of argument spans.

Sense embeddings can serve as an entry point to many other knowledge bases linked to WordNet, such as the multilingual knowledge graph of BabelNet [91], the commonsense triples of ConceptNet [68] or WebChild [92], the semantic frames of VerbNet [93], and even the images of ImageNet [94] or Visual Genome [95]. Recent works have used the symbolic relations expressed in these knowledge bases to improve neural solutions to Natural Language Inference [96], Commonsense Reasoning [97], Story Generation [98], among others. As an example of how using LMMS-SP to bridge natural language and symbolic knowledge can be beneficial, in Figure 2.6, we demonstrate how sense embeddings allow for the generalization of argument spans, predicted by a semantic parser, exploiting WordNet relations between matched synsets.

2.4 Probing NLMs for Senses and Commonsense

Layer-wise Sense Probing

Understanding properties about the internal states of NLMs has become an important line of research known as 'model probing'. Probing operates under the assumption that if a relatively simple classifier, based exclusively on representations from NLMs, performs well at some non-trivial task, that shows the information required for that task was already encoded in those representations. Generically, probes are defined as functions (learned or heuristic) designed to reveal some intrinsic property of NLMs. Loureiro et al. [28] provides an extended introduction covering several NLM probing methodologies.

The main contribution of Loureiro et al. [28] is a principled approach for sense representation, featuring a better supported alternative to the sum of a particular number of top layers, as done in [26] (following Devlin et al. [25]). This improved approach is based on probing contextual embeddings from each layer composing NLMs, and using the resulting analysis to inform a weighted pooling operation combining contextual embeddings from all layers. This approach allows us to determine layer and model specific weights specifically tuned for sense representation. These weights are normalized using softmax with a temperature parameter t [99] which is only specific to Sense Profiles, not NLMs (and determined empirically).

Using a custom validation set proposed specifically for probing sense representations (i.e., does not require use glosses or propagation), we find that different NLMs* of similar architecture exhibit best performance at different layers (see Table 2.1). These results support prior work showing that bottom layers do not exhibit sufficient context-specificity for disambiguation tasks [100]. Most interestingly, our results provide additional evidence that top-most layers also may not be the best suited for lexical semantics [101, 102], supporting more nuanced explanations for this variation, such as the Information-Bottleneck hypothesis of Voita et al. [103]. This irregular variation is most clear when comparing the distribution of the best layers for XLNet and the other NLMs shown in Table 2.1. Considering silhouette scores† [104] and PCA visualizations of the embedding space (see Figure 2.7), we arrive at similar conclusions, namely that final layers tend to produce less accurate representations than layers closer to the middle, while the first layer show lowest scores (i.e., worst clustering).

*Loureiro et al. [28] reports layer-wise probing analyses for up to 14 NLM variants.
†Silhouette coefficients are based on intra- and nearest-cluster cosine similarities.

	INIT	-24	-23	-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
BERT-L	53	58	62	63	65	67	68	68	69	70	71	71	71	71	72	72	71	72	73	72	73	74	**75**	**75**	72
XLNet-L	51	57	65	67	68	70	71	72	**73**	**73**	72	72	72	72	71	71	71	71	71	71	72	72	72	72	68
RoBERTa-L	53	57	63	66	67	69	71	72	73	73	74	74	74	74	**75**	**75**	**75**	74	**75**	74	74	74	73	74	71
ALBERT-XL	54	65	67	68	69	70	70	**71**	**71**	**71**	**71**	**71**	**71**	**71**	**71**	70	70	69	69	69	69	69	69	69	64

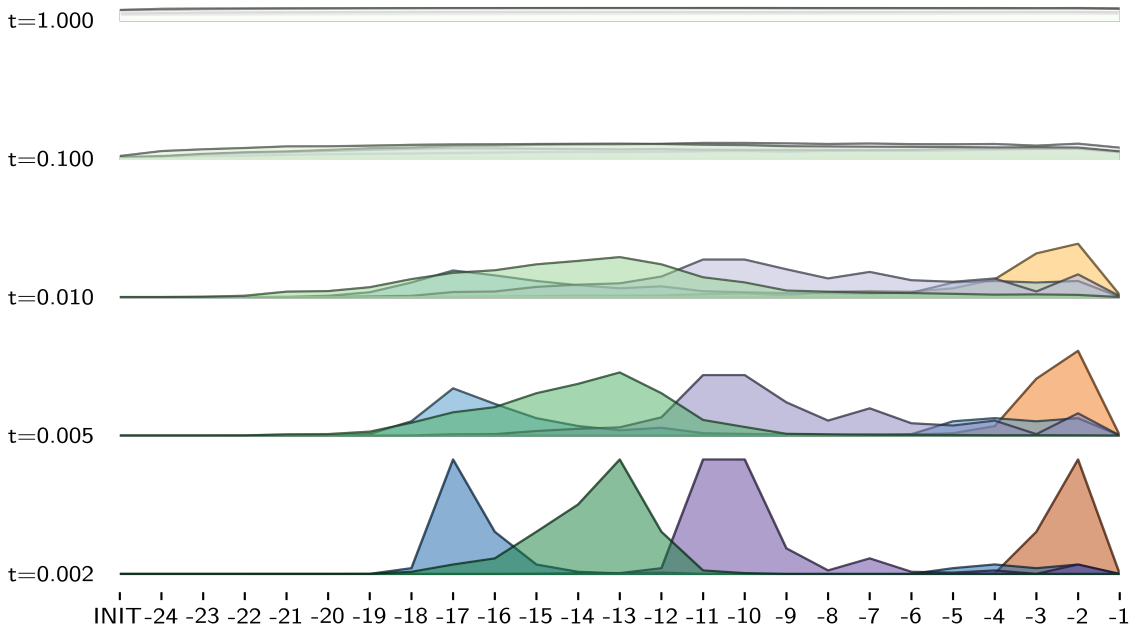


TABLE 2.1: Shows interaction between F1 scores (rounded) for 1-NN WSD using each layer of four different NLMs, and respective weight distributions (matching colors) using decreasing temperature (t) parameters. Lower temperatures induce higher skewness towards layers that perform best on the probing validation set. Distributions based on t=1.000 are almost uniform, while t <0.002 places almost all mass on single best layer.

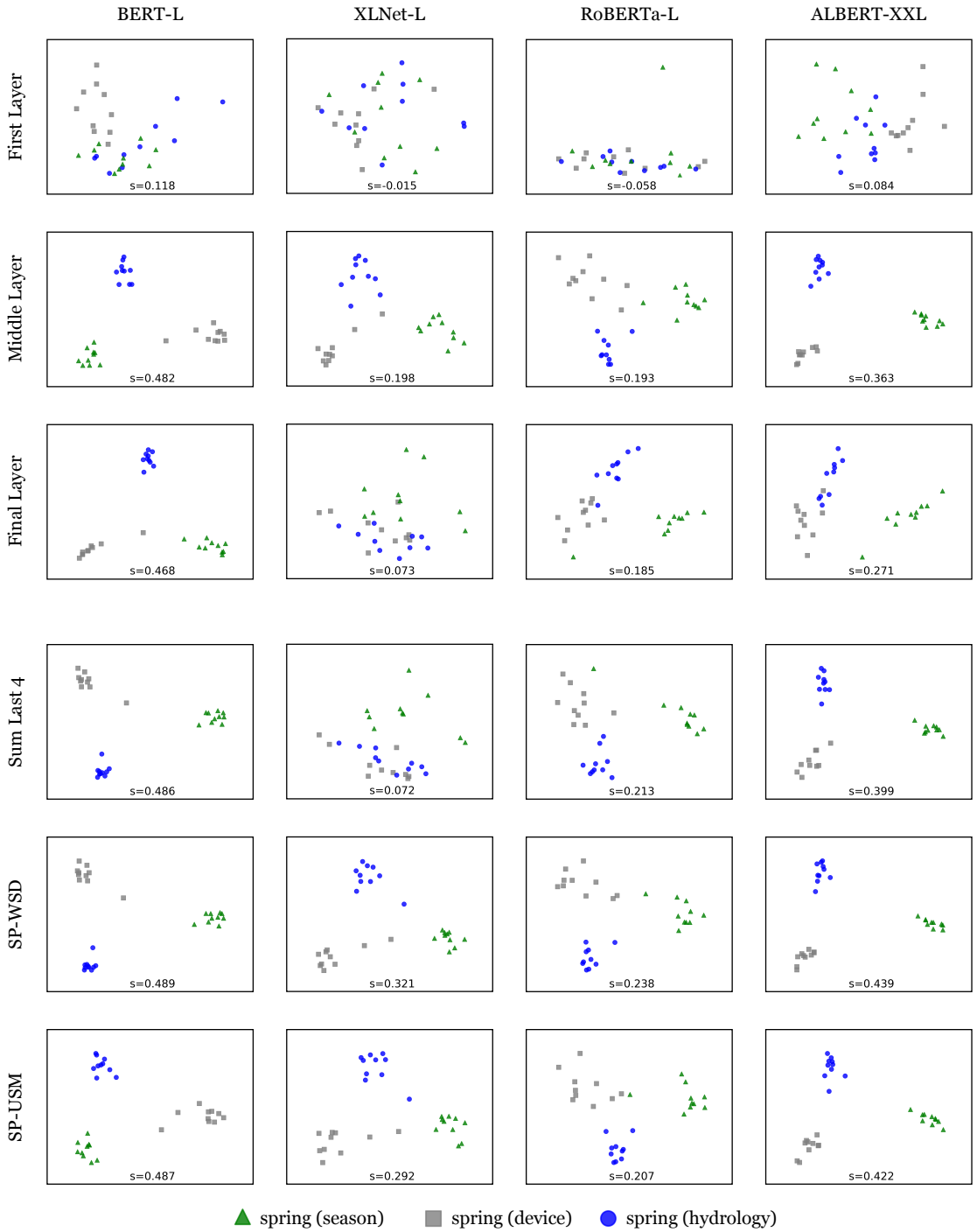


FIGURE 2.7: Visualization of embedding spaces using different pooling strategies, using PCA for dimensionality reduction. Each point corresponds to an embedding for the word 'spring' in context, from the 10-shot set of CoarseWSD-20 [30]. Silhouette scores s are computed before reduction (higher is better). Bottom two rows correspond to our proposed pooling strategies, showing the best clustering out of all reported options.

Grounded Commonsense Knowledge

	Core (4,960 candidates)					Full Inventory (117,659 candidates)					
	P@1	**P@3**	**P@10**	**P@100**	**MRR**	**P@1**	**P@3**	**P@10**	**P@100**	**P@1000**	**MRR**
All	24.41	40.56	59.10	83.20	35.64	7.18	13.78	23.09	45.75	71.75	12.55
WordNet	31.25	49.80	69.10	87.82	43.46	7.78	14.75	24.26	46.39	71.84	13.34
Hypernym	29.04	45.96	66.15	86.10	40.77	8.31	17.24	30.77	59.17	82.74	15.65
Holonym (Member)	42.31	69.23	88.46	100.00	57.80	1.75	3.04	5.03	13.98	41.89	3.00
Holonym (Part)	34.48	60.69	80.69	92.41	50.20	13.97	25.93	40.63	67.89	88.15	22.91
Antonym	37.94	58.16	74.11	91.49	50.09	8.10	13.72	20.96	40.56	70.32	12.55
Meronym (Substance)	43.75	81.25	81.25	100.00	59.14	2.43	6.23	12.46	33.13	65.50	6.00
WikiData	16.18	33.09	49.26	79.41	27.62	5.05	10.12	18.83	43.91	72.07	9.69
P31 (Instance of)	10.26	23.08	23.08	61.54	16.94	2.90	6.74	13.61	37.77	68.56	6.67
P361 (Part of)	15.56	35.56	62.22	82.22	30.26	8.71	16.17	27.21	55.89	79.37	14.86
P366 (Use)	14.81	25.93	48.15	88.89	24.80	4.06	9.70	19.27	42.07	64.74	9.00
P186 (Made from)	33.33	46.67	60.00	86.67	41.66	8.61	12.83	23.63	46.64	72.77	13.03
P461 (Opposite of)	20.00	60.00	80.00	100.00	43.91	8.98	18.36	30.34	60.28	81.24	16.35
ConceptNet	13.86	25.87	43.51	75.78	23.38	4.55	9.88	18.07	42.11	70.06	9.23
AtLocation	14.02	25.91	46.95	79.27	24.24	4.98	10.56	19.82	45.82	76.10	10.09
UsedFor	7.41	16.67	36.42	75.93	16.04	3.18	8.17	15.13	38.88	69.59	7.48
IsA	27.50	43.33	62.50	87.50	38.56	7.42	13.67	27.34	59.38	83.59	13.61
Causes	5.26	23.68	34.21	65.79	16.56	2.68	6.25	12.05	27.68	54.91	5.84
HasSubevent	3.51	14.04	19.30	43.86	10.08	0.98	2.44	5.37	14.63	35.12	2.64
HasPrerequisite	4.00	16.00	26.00	78.00	13.16	3.64	8.48	13.94	41.21	71.52	7.35
HasProperty	4.26	14.89	38.30	76.60	14.65	2.55	5.10	9.55	29.30	63.69	5.21
CapableOf	8.33	18.75	33.33	54.17	16.09	2.44	7.32	13.82	30.08	51.22	6.48
MotivatedByGoal	29.73	51.35	67.57	89.19	43.59	6.73	20.19	29.81	63.46	80.77	15.66

TABLE 2.2: Results on the SenseLAMA using LMMS-SP$_{\text{BERT-L}}$ embeddings (SP-USM, synset-level), for most frequent relations. Sorted by P@1 on Full Inventory results.

In Section 2.3 we describe how we perform sense-level relation extraction grounded in WordNet using a BERT model augmented with explicit sense-level representations (i.e., SynBERT), according to Loureiro and Jorge [33]. Evaluating the performance of SynBERT on the SenseLAMA probing task also introduced in Loureiro and Jorge [33], we obtain the results reported in Table 2.2.

Considering only instances targeting core synsets (i.e., frequent concepts), we find P@10 above 30% for most relations, and over 80% for relations such as *Holonym (Part)*, suggesting that extraction for some relation types could be reliable enough for some applications. Admitting instances targeting any synset (Full Inventory) we find much lower results, which is to be expected considering the 20x increase for the search space. Nevertheless, we still find that most relations can be accurately predicted from the top 1% of candidates ($\leq$ 60% P@1000). Most notably, these results support that commonsense relations are harder to model by NLMs than lexical or encyclopedic relations.

2.5 Evaluation and State-of-the-Art

We track the quality of our sense representations by their performance on the various sense-related tasks described in Section 2.1. Out of those 6 tasks, WSD stands out as the one with most interest from the NLP commmunity, and most related works.

With the introduction of LMMS [26], we raised the state-of-the-art for WSD by an unprecedented 4.6 F1 on the *de facto* evaluation set of Raganato et al. [75], but that incidental result (of 75.4 F1), from our work focused on sense representation, would be quickly surpassed. In Loureiro et al. [30] we provide an extensive overview of different WSD solutions, highlighting BEM [105] and EWISER [82] as the best performing systems, achieving a performance of 79.0 F1 and 80.1 F1 on [75], correspondingly. Shortly after, in Loureiro et al. [28] we report a new result from ConSeC [106] achieving a remarkable 82.3 F1 result on the same test set, surpassing estimated human performance [107]. The improvements we propose with LMMS-SP [28] mostly impact performance on the other tasks.

Word-in-Context (WiC) is another task with relevant related work. During the Shared Task competition that ran shortly after the release of WiC, our system based on LMMS ranked second place [29] with an accuracy of 68.01%, behind a fine-tuning system that obtained 68.36%. Since then, WiC has become part of the most popular benchmark for NLMs, SuperGLUE [73]. As a consequence of this, several of the latest NLMs have reported performance on WiC, with the current best result achieving 77.9% accuracy [108], near the estimated human performance of 80% accuracy.

However, all of these related works are using supervised systems, either fine-tuning NLMs or using them to train additional classifiers, so these results are not directly comparable to ours. Besides fine-tuned systems not being aligned with our research goals, in Loureiro et al. [30] we also demonstrate that, in spite of generally lower performance, 1-NN offers several advantages over fine-tuning, most notably, better performance in few-shot settings (see Figure 2.8). In Rezaee et al. [31] we also show how sense embeddings may be used with multilingual NLMs for cross-lingual WSD.

Strictly focused on learning sense representations from NLMs, relying on 1-NN for classification, we find fewer related works, with SensEmBERT [81] and ARES [83] being the only that we are aware of (both are based on BERT-Large). SensEmBERT and ARES also report performance on WiC, but they use their sense embeddings as part of fine-tuning, so those results reported from their publications are also not directly comparable. There are also the recently introduced LessLex [109] sense embeddings based on an

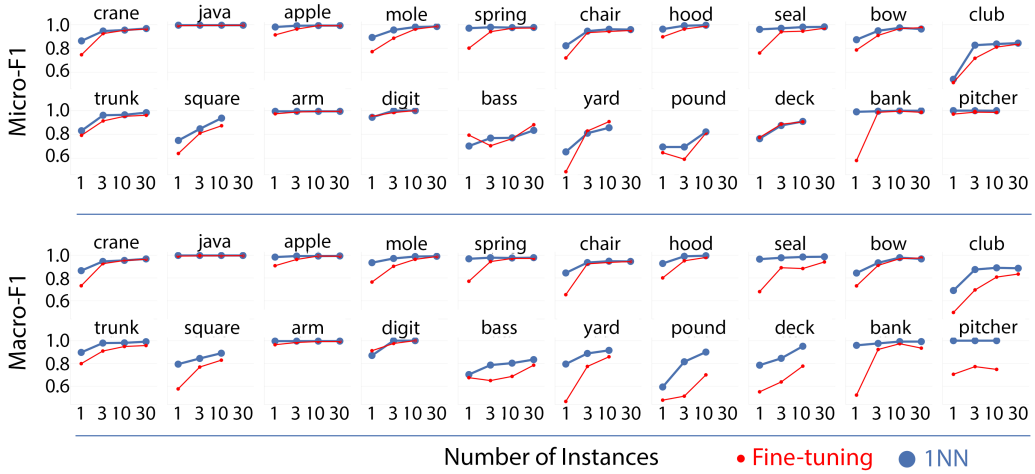


FIGURE 2.8: Micro and macro F1 on the CoarseWSD-20 dataset [30] for different sizes of n in the n-shot setting, comparing 1-NN with fine-tuning strategies for WSD.

ensemble of static embeddings mapped to BabelNet. Still, out of these three recent sets of sense embeddings, only ARES reaches full-coverage of WordNet, making those sense embeddings the most comparable to ours. From before the development of contextual embeddings, we highlight DeConf [76] as another full-coverage set of sense embeddings. Since DeConf is based on static embeddings, it cannot be used for 1-NN in the latent space of NLMs, unlike our sense embeddings or ARES.

An overview of LMMS-SP results using various NLMs can found on Table 2.3. As explained earlier, this table only reports results in comparison with ARES [83] and DeConf [23] since these are the two related works with sets of sense embeddings most directly comparable to ours.

Embeddings	**WSD (F1)**	**USM (P@5)**	**WiC (ACC)**	**GWSC (COR)**	**SCWS (COR)**	**SID (COR)**
DeConf [23]	N/A	N/A	58.7†	N/A	71.5†	75.1
ARES [83]	**77.9**	84.7	67.6	76.9	67.9	70.6
LMMS-SP$_{\text{BERT-L}}$	75.2	86.7	67.4	76.3	64.1	77.8
LMMS-SP$_{\text{XLNet-L}}$	74.1	87.3	66.1	**78.7**	**75.9**	**79.5**
LMMS-SP$_{\text{RoBERTa-L}}$	75.2	86.9	67.8	75.7	67.4	74.1
LMMS-SP$_{\text{ALBERT-XXL}}$	75.5	**87.6**	**67.9**	75.2	69.9	77.4

TABLE 2.3: Comparison between LMMS-SP, using different NLMs, and related works with full-coverage sense embeddings. DeConf represent sense embeddings without using NLMs, thus cannot be used (N/A) with our 1-NN approach and applied to WSD, USM, or GWSC (†for completeness, we report results for SCWS obtained from other works, based on different approaches). Reports results from the development set of WiC.

Considering that USM is the most relevant task for our goal of using NLMs to match commonsense concepts (see Section 1.1), in Table 2.4 we also report results focused on that task, highlighting the progress achieved from successive refinements over various works. The introduction of the UWA corpus [27] provided the largest improvement, while the principled approach of LMMS-SP [28] allowed for further gains, specially in P@5.

Sense Embeddings		**F1**	**P@5**	**MRR**
Ours [26] (Jul. 2019)	LMMS	52.2	66.9	59.0
Ours [27] (Nov. 2020)	$\text{LMMS}_{\text{BERT-L}}$ w/UWA	54.9	74.1	63.5
	$\text{LMMS}_{\text{RoBERTa-L}}$ w/UWA	62.1	80.2	70.1
Scarlini et al. [83] (Nov. 2020)	ARES (BERT-L)	61.4	84.7	71.8
Ours [28] (Jan. 2022)	$\text{LMMS-SP}_{\text{BERT-L}}$	60.8	86.7	72.2
	$\text{LMMS-SP}_{\text{XLNet-L}}$	60.1	87.3	71.9
	$\text{LMMS-SP}_{\text{RoBERTa-L}}$	62.2	86.9	73.1
	$\text{LMMS-SP}_{\text{ALBERT-XXL}}$	**62.9**	**87.6**	**73.7**

TABLE 2.4: Showing progress on the USM task using our sense embeddings improved over successive publications, and related works. Ordered by publication date.

Finally, in Table 2.5 we report results on the SID task, showing how our approach produces sense representations which are much more highly correlated with human similarity judgements than prior works, with the exception of LessLex (an ensemble approach).

	Sense Embeddings	**WN Full Coverage**	**COR (n=354)**
Static	fastText [28, 51]	✓	63.5
	$\text{NASARI}_{\text{UMBC}}$ [22]		71.6
	DeConf [23]	✓	74.9
	LessLex [109]		**82.3**
Contextual	SensEmBERT [81]		66.8
	ARES [83]	✓	70.4
	LMMS [26]	✓	72.2
	$\text{LMMS-SP}_{\text{BERT-L}}$ [28]	✓	77.8
	$\text{LMMS-SP}_{\text{XLNet-L}}$ [28]	✓	**79.6**
	$\text{LMMS-SP}_{\text{RoBERTa-L}}$ [28]	✓	74.2
	$\text{LMMS-SP}_{\text{ALBERT-XXL}}$ [28]	✓	77.2

TABLE 2.5: Performance (Pearson Correlation) on the overlapping subset of the SID dataset. All reported embeddings feature 300 dimensions (reduced to this dimensionality using SVD where applicable). LessLex and NASARI were converted from BabelNet to WordNet using the same mapping applied to adaptation of the SID dataset.

Chapter 3

Discussion

While we believe to have been successful in pursuing our research objectives as stated in Section 1.1, the initial plan for this thesis included exploring the application of our proposed sense representations in PL systems, such as ProbLog [110], using standardized inference mechanisms. Inspired by prior attempts at combining distributional representations with First-Order-Logic (FOL) [15], we planned on using Natural Language Inference (NLI) tasks to explore how our sense representations, learned from the recently emerging NLMs, could result in performance improvements and more interpretable solutions. For NLI tasks*, systems are provided with premise and hypothesis sentence pairs (e.g., P: "Bob is sleeping."; H: "Bob is eating."), and need to predict whether the hypothesis is implied (entailment), in contradiction, or irrelevant (neutral) to the premise. This format used for NLI allows a PL system to arrive at predictions based on difference between prior and posterior probabilities of facts extracted from hypotheses, which would vary according to the facts extracted from premises provided as evidence, among other possible solutions. Below, we explain our setbacks with that initial plan.

The most significant hurdle was the fact that these recent NLMs are, in fact, a profound paradigm shift for NLP, and consequently what has become expected of state-of-the-art solutions in this field. From ELMo [55] in 2018 (our start date), to BERT [25] in 2019, T5 [112] and GPT-3 [2] in 2020, and now PaLM [7] in 2022, each of these NLMs has managed to significantly improve performance on key NLP tasks over the previous. Consequently, challenging benchmarks quickly became outdated [113, 114], along with test sets targeting limitations of a particular generation of NLMs [115, 116]. This remarkable progress has, however, dramatically reduced the opportunity for improvements from complementary

*Considering Recognizing Textual Entailment (RTE) [111] tasks to be a subset of NLI.

approaches, such as neurosymbolic solutions. Previously hard subsets of NLI test sets are now correctly labeled by the latest NLMs, and the instances for which they fail, while certainly interesting, are also not likely to become trivial for complementary approaches. As it stands, a neurosymbolic system must have near perfect understanding of the concepts involved in the contexts of premises and hypothesis of NLI instances, along with near perfect understanding of the predicate-argument structures relating those concepts. Although we did spend considerable time* researching predicate-argument structures based on Semantic Role Labeling (SRL), we found that addressing these two requirements goes beyond the scope our doctoral thesis, and decided to focus on the representation and matching of commonsense concepts using the best NLMs available to us.

Initial experiments with ProbLog also revealed that it would be challenging to use a large Knowledge Base (specific to CSK) as background knowledge for inference, assuming we would be able to build one following automated construction methods. Not only are inference runtimes significantly impacted by the size of this Knowledge Base, but the relational representation format is also not trivial. The number of arguments used with rules or relations should result from an optimal balance between the expressiveness of different roles in SRL, and the tractability limitations of inference systems. If rules need to discovered from rule-mining solutions such as AMIE [119], as we also considered, then the number of arguments is in opposition to tractability once again.

Some very recent works have managed to report relevant progress towards logic-based hybrid approaches for NLI. Most notably, Stacey et al. [120] has introduced a framework applying handcrafted logical rules at the span-level, which is conceivably the first logic-based system to rival the performance of fine-tuned NLMs, although this solution is not based on FOL, or even formal logical inference. Before this result, neurosymbolic solutions have usually been limited to simpler subsets of NLI [121] (e.g., monotonicity), or selectively choose (based on a learned classifier) which instances they resolve using deep learning or symbolic logic [122]. Other tasks that have also been recently used to explore neurosymbolic solutions are Question Answering [123] and Link Prediction [124].

We also note that commonly held beliefs about the limitations of NLMs for planning and reasoning are currently being challenged by recent findings [125, 126].

*Leading to embeddings specialized on affordance [117], and findings on cross-lingual transfer [118].

Chapter 4

Conclusions

In this thesis we have shown that it is possible to produce sense embeddings applicable beyond disambiguation, with relevant implications for long-standing challenges in Artificial Intelligence, such as symbol grounding [127].

With LMMS [26], and its iterations [27–29, 31], we propose a principled approach for learning distributional representations of word senses from pre-trained NLMs, focusing on state-of-the-art Transformer models. From extensive evaluation on several sense-related tasks, we demonstrated that our proposed approach is more effective than prior work at approximating precise word sense representations in the same vector space of NLMs [23, 81, 83]. The broad probing analysis of the many variants of popular NLMs targeted in the scope of this work provides new evidence supporting further research on the interplay between pre-training objectives, layer specialization, and model size.

Although improving the state-of-the-art for WSD is not among our research objectives, we have managed to achieve such results on occasion [26], besides proposing various improvements to the evaluation and analysis of WSD systems, particularly on the comparison between similarity (i.e., 1-NN) and fine-tuning approaches [30]. Incidentally, while entity linking in the medical domain is also outside our research objectives, our work on MedLinker [32] presents a competitive system for that task, and demonstrates how similarity from embedding approaches can be combined with dictionary matching.

Finally, with SynBERT [33], we show that sense embeddings, learned from grounded ontologies, can be integrated into pre-trained NLMs, allowing for a more precise and extensive probing of commonsense knowledge learned during pre-training compared to prior work such as LAMA [70]. We also explore how SynBERT, or similar models, can be used to extract novel commonsense knowledge graphs which may support recent hybrid

methods fusing knowledge graphs with NLMs [128], or enable symbolic-first methods [17] to leverage precise CSK learned without supervision by NLMs.

Effectively, there are known limitations to meaning representation based on language modelling objectives alone [129, 130], and there is still much to understand about how to best leverage NLMs for meaning representation. Nonetheless, we believe our work presents strong evidence supporting the feasibility of using NLMs for symbolic concept representation, following grounded ontologies. In turn, we hope our research contributes towards improved commonsense reasoning by symbolic manipulation with neurosymbolic hybrids, and the next set of breakthroughs for more powerful, and interpretable, NLP systems [131].

Bibliography

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf [Cited on page 1.]

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165 [Cited on pages 1, 7, and 27.]

[3] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 610–623. [Online]. Available: https://doi.org/10.1145/3442188.3445922 [Cited on page 1.]

[4] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022. [Online]. Available: https://arxiv.org/abs/2203.02155

[5] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in*

Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf [Cited on page 1.]

[6] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020. [Online]. Available: https://arxiv.org/abs/2001.08361 [Cited on page 1.]

[7] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "Palm: Scaling language modeling with pathways," 2022. [Online]. Available: https://arxiv.org/abs/2204.02311 [Cited on page 27.]

[8] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," 2022. [Online]. Available: https://arxiv.org/abs/2206.07682 [Cited on page 1.]

[9] H. Zhang, L. H. Li, T. Meng, K.-W. Chang, and G. V. d. Broeck, "On the paradox of learning to reason from data," 2022. [Online]. Available: https://arxiv.org/abs/2205.11502 [Cited on page 1.]

[10] P. Domingos and W. Webb, "A tractable first-order probabilistic logic," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, pp. 1902–1909, Sep. 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/8398 [Cited on page 1.]

[11] G. Lakemeyer and H. J. Levesque, "A tractable, expressive, and eventually complete first-order logic of limited belief," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 1764–1771. [Online]. Available: https://doi.org/10.24963/ijcai.2019/244 [Cited on page 1.]

[12] A. S. d'Avila Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran, "Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning," *FLAP*, vol. 6, no. 4, pp. 611–632, 2019. [Online]. Available: https://collegepublications.co.uk/ifcolog/?00033 [Cited on page 1.]

[13] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence: Current trends," 2021. [Online]. Available: https://arxiv.org/abs/2105.05330 [Cited on page 1.]

[14] T. A. Poggio and F. Anselmi, "Visual cortex and deep networks: Learning invariant representations," 2016. [Cited on page 1.]

[15] I. K. A. Beltagy, "Natural language semantics using probabilistic logic," Ph.D. dissertation, 2016. [Cited on pages 2 and 27.]

[16] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "QA-GNN: Reasoning with language models and knowledge graphs for question answering," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 535–546. [Online]. Available: https://aclanthology.org/2021.naacl-main.45 [Cited on pages 2 and 9.]

[17] J. Huang, Z. Li, B. Chen, K. Samel, M. Naik, L. Song, and X. Si, "Scallop: From probabilistic deductive databases to scalable differentiable reasoning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 25 134–25 145. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/d367eef13f90793bd8121e2f675f0dc2-Paper.pdf [Cited on pages 2 and 30.]

[18] J. Camacho-Collados and M. T. Pilehvar, "From word to sense embeddings: A survey on vector representations of meaning," *J. Artif. Int. Res.*, vol. 63, no. 1,

pp. 743–788, Sep. 2018. [Online]. Available: https://doi.org/10.1613/jair.1.11259 [Cited on pages 2 and 6.]

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, p. 3111–3119. [Cited on page 2.]

[20] S. Rothe and H. Schütze, "AutoExtend: Extending word embeddings to embeddings for synsets and lexemes," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 1793–1803. [Online]. Available: https://aclanthology.org/P15-1173 [Cited on pages 2 and 6.]

[21] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "SensEmbed: Learning sense embeddings for word and relational similarity," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 95–105. [Online]. Available: https://aclanthology.org/P15-1010

[22] J. Camacho-Collados, M. T. Pilehvar, and R. Navigli, "NASARI: a novel approach to a semantically-aware representation of items," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 567–577. [Online]. Available: https://aclanthology.org/N15-1059 [Cited on page 26.]

[23] M. T. Pilehvar and N. Collier, "De-conflated semantic representations," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1680–1690. [Online]. Available: https://aclanthology.org/D16-1174 [Cited on pages 2, 6, 25, 26, and 29.]

[24] J. M. Rodd, "Settling into semantic space: An ambiguity-focused account of word-meaning access," *Perspectives on Psychological Science*, vol. 15, no. 2, pp. 411–427, 2020, pMID: 31961780. [Online]. Available: https://doi.org/10.1177/1745691619885860 [Cited on page 2.]

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423 [Cited on pages 2, 7, 12, 20, and 27.]

[26] D. Loureiro and A. Jorge, "Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5682–5691. [Online]. Available: https://aclanthology.org/P19-1569 [Cited on pages 3, 4, 5, 10, 12, 14, 15, 20, 24, 26, and 29.]

[27] D. Loureiro and J. Camacho-Collados, "Don't neglect the obvious: On the role of unambiguous words in word sense disambiguation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 3514–3520. [Online]. Available: https://aclanthology.org/2020.emnlp-main.283 [Cited on pages 4, 5, 12, 14, 26, and 29.]

[28] D. Loureiro, A. Mário Jorge, and J. Camacho-Collados, "LMMS reloaded: Transformer-based sense embeddings for disambiguation and beyond," *Artificial Intelligence*, vol. 305, p. 103661, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370222000017 [Cited on pages 3, 4, 5, 10, 12, 15, 17, 20, 24, and 26.]

[29] D. Loureiro and A. Jorge, "LIAAD at SemDeep-5 challenge: Word-in-context (WiC)," in *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*. Macau, China: Association for Computational Linguistics, Aug. 2019, pp. 1–5.

[Online]. Available: https://aclanthology.org/W19-5801 [Cited on pages 3, 4, 5, 24, and 29.]

[30] D. Loureiro, K. Rezaee, M. T. Pilehvar, and J. Camacho-Collados, "Analysis and Evaluation of Language Models for Word Sense Disambiguation," *Computational Linguistics*, vol. 47, no. 2, pp. 387–443, 07 2021. [Online]. Available: https://doi.org/10.1162/coli_a_00405 [Cited on pages 4, 22, 24, 25, and 29.]

[31] K. Rezaee, D. Loureiro, J. Camacho-Collados, and M. T. Pilehvar, "On the cross-lingual transferability of contextualized sense embeddings," in *Proceedings of the 1st Workshop on Multilingual Representation Learning*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 107–115. [Online]. Available: https://aclanthology.org/2021.mrl-1.10 [Cited on pages 3, 4, 24, and 29.]

[32] D. Loureiro and A. M. Jorge, "MedLinker: Medical Entity Linking with Neural Representations and Dictionary Matching," in *Advances in Information Retrieval*, J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, and F. Martins, Eds. Cham: Springer International Publishing, 2020, pp. 230–237. [Cited on pages 3, 4, 5, 17, and 29.]

[33] ——, "Precisely Probing Commonsense Knowledge in Pretrained Language Models using Sense Embeddings," in *Under Review*, sep 2022. [Cited on pages 3, 4, 5, 15, 18, 23, and 29.]

[34] J. R. Firth, "The technique of semantics." *Transactions of the Philological Society*, vol. 34, no. 1, pp. 36–73, 1935. [Cited on page 6.]

[35] L. Wittgenstein, "Philosophical investigations, trans," *GEM Anscombe*, vol. 261, p. 49, 1953. [Cited on page 6.]

[36] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954. [Cited on page 6.]

[37] G. Salton, "The smart system," *Retrieval Results and Future Plans*, 1971. [Cited on page 6.]

[38] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[39] S. C. Deerwester, S. T. Dumais, G. W. Furnas, R. A. Harshman, T. K. Landauer, K. E. Lochbaum, and L. A. Streeter, "Computer information retrieval using latent semantic structure," Jun. 13 1989, uS Patent 4,839,853.

[40] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[41] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior research methods, instruments, & computers*, vol. 28, no. 2, pp. 203–208, 1996.

[42] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological review*, vol. 104, no. 2, p. 211, 1997. [Cited on page 6.]

[43] H. Schutze, "Dimensions of meaning," in *Supercomputing'92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*. IEEE, 1992, pp. 787–796. [Cited on page 6.]

[44] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, Jun. 1995, pp. 189–196. [Online]. Available: https://aclanthology.org/P95-1026 [Cited on page 6.]

[45] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003. [Cited on page 6.]

[46] R. Collobert and J. Weston, "Fast semantic extraction using a novel neural network architecture," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 560–567. [Online]. Available: https://aclanthology.org/P07-1071 [Cited on page 6.]

[47] ——, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[48] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011. [Cited on page 6.]

[49] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119. [Cited on page 6.]

[50] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162 [Cited on page 6.]

[51] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017. [Online]. Available: https://aclanthology.org/Q17-1010 [Cited on pages 6 and 26.]

[52] J. Camacho-Collados, M. T. Pilehvar, and R. Navigli, "Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities," *Artificial Intelligence*, vol. 240, pp. 36 – 64, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370216300820 [Cited on page 6.]

[53] B. Athiwaratkun, A. Wilson, and A. Anandkumar, "Probabilistic FastText for multi-sense word embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1–11. [Online]. Available: https://aclanthology.org/P18-1001 [Cited on page 6.]

[54] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning generic context embedding with bidirectional LSTM," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 51–61. [Online]. Available: https://aclanthology.org/K16-1006 [Cited on page 7.]

[55] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: https://aclanthology.org/N18-1202 [Cited on pages 7 and 27.]

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. [Cited on page 7.]

[57] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in BERTology: What we know about how BERT works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.54 [Cited on page 7.]

[58] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27. [Cited on page 7.]

[59] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2978–2988. [Online]. Available: https://aclanthology.org/P19-1285 [Cited on page 7.]

[60] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, 2019, pp. 5753–5763. [Cited on page 7.]

[61] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012 [Cited on page 8.]

[62] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692 [Cited on page 8.]

[63] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Cited on page 8.]

[64] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS [Cited on page 8.]

[65] C. Fellbaum, "Wordnet : an electronic lexical database." MIT Press, May 1998. [Cited on page 9.]

[66] H. Liu and P. Singh, "Conceptnet — a practical commonsense reasoning tool-kit," *BT Technology Journal*, vol. 22, pp. 211–226, 2004. [Cited on page 9.]

[67] J. Gordon and B. V. Durme, "Reporting bias and knowledge acquisition," in *AKBC '13*, 2013. [Cited on page 9.]

[68] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17. AAAI Press, 2017, p. 4444–4451. [Cited on pages 9, 18, and 19.]

[69] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4149–4158. [Online]. Available: https://aclanthology.org/N19-1421 [Cited on page 9.]

[70] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China:

Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. [Online]. Available: https://aclanthology.org/D19-1250 [Cited on pages 9, 18, and 29.]

[71] S. Jastrzebski, D. Bahdanau, S. Hosseini, M. Noukhovitch, Y. Bengio, and J. Cheung, "Commonsense mining as knowledge base completion? a study on the impact of novelty," in *Proceedings of the Workshop on Generalization in the Age of Deep Learning*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 8–16. [Online]. Available: https://aclanthology.org/W18-1002 [Cited on page 9.]

[72] X. Li, A. Taheri, L. Tu, and K. Gimpel, "Commonsense knowledge base completion," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1445–1455. [Online]. Available: https://aclanthology.org/P16-1137 [Cited on page 9.]

[73] P. Wang, N. Peng, F. Ilievski, P. Szekely, and X. Ren, "Connecting the dots: A knowledgeable path generator for commonsense question answering," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4129–4140. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.369 [Cited on pages 9 and 24.]

[74] R. Navigli, "Word sense disambiguation: A survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 10:1–10:69, Feb. 2009. [Online]. Available: http://doi.acm.org/10.1145/1459352.1459355 [Cited on page 10.]

[75] A. Raganato, J. Camacho-Collados, and R. Navigli, "Word sense disambiguation: A unified evaluation framework and empirical comparison," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 99–110. [Online]. Available: https://aclanthology.org/E17-1010 [Cited on pages 10 and 24.]

[76] M. T. Pilehvar and J. Camacho-Collados, "WiC: the word-in-context dataset for evaluating context-sensitive meaning representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 1267–1273.

[Online]. Available: https://aclanthology.org/N19-1128 [Cited on pages 11, 17, and 25.]

[77] C. S. Armendariz, M. Purver, S. Pollak, N. Ljubešić, M. Ulčar, I. Vulić, and M. T. Pilehvar, "SemEval-2020 task 3: Graded word similarity in context," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 36–49. [Online]. Available: https://aclanthology.org/2020.semeval-1.3 [Cited on pages 11 and 17.]

[78] E. Huang, R. Socher, C. Manning, and A. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 873–882. [Online]. Available: https://aclanthology.org/P12-1092 [Cited on pages 11 and 17.]

[79] D. Colla, E. Mensa, and D. P. Radicioni, "Sense identification data: A dataset for lexical semantics," *Data in Brief*, vol. 32, p. 106267, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340920311616 [Cited on page 11.]

[80] J. Camacho-Collados, M. T. Pilehvar, N. Collier, and R. Navigli, "SemEval-2017 task 2: Multilingual and cross-lingual semantic word similarity," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 15–26. [Online]. Available: https://aclanthology.org/S17-2002 [Cited on page 11.]

[81] B. Scarlini, T. Pasini, and R. Navigli, "SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation," in *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence, 2020, pp. 8758–8765. [Cited on pages 12, 24, 26, and 29.]

[82] M. Bevilacqua and R. Navigli, "Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul.

2020, pp. 2854–2864. [Online]. Available: https://aclanthology.org/2020.acl-main.255 [Cited on page 24.]

[83] B. Scarlini, T. Pasini, and R. Navigli, "With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 3528–3539. [Online]. Available: https://aclanthology.org/2020.emnlp-main.285 [Cited on pages 24, 25, 26, and 29.]

[84] G. Berend, "Sparsity makes sense: Word sense disambiguation using sparse contextualized word representations," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 8498–8508. [Online]. Available: https://aclanthology.org/2020.emnlp-main.683 [Cited on page 12.]

[85] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas, "Using a semantic concordance for sense identification," in *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. [Online]. Available: https://www.aclweb.org/anthology/H94-1046 [Cited on page 12.]

[86] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," in *Journal of machine learning research*, vol. 9, no. Nov, 2008, pp. 2579–2605. [Cited on page 14.]

[87] D. Loureiro and A. Jorge, "LIAAD at SemDeep-5 challenge: Word-in-context (WiC)," in *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*. Macau, China: Association for Computational Linguistics, Aug. 2019, pp. 1–5. [Online]. Available: https://aclanthology.org/W19-5801 [Cited on page 17.]

[88] N. Okazaki and J. Tsujii, "Simple and efficient algorithm for approximate dictionary matching," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 851–859. [Online]. Available: https://aclanthology.org/C10-1096 [Cited on page 17.]

[89] P. Dufter, N. Kassner, and H. Schütze, "Static embeddings as efficient knowledge bases?" in *Proceedings of the 2021 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics, Jun. 2021, pp. 2353–2363. [Online]. Available: https://aclanthology.org/2021.naacl-main.186 [Cited on page 18.]

[90] V. Swamy, A. Romanou, and M. Jaggi, "Interpreting language models through knowledge graph extraction," in *eXplainable AI approaches for debugging and diagnosis.*, 2021. [Online]. Available: https://openreview.net/forum?id=PW4AGjla3sx [Cited on page 18.]

[91] R. Navigli and S. P. Ponzetto, in *BabelNet: Building a Very Large Multilingual Semantic Network*, 2010, pp. 216–225. [Cited on page 19.]

[92] N. Tandon, G. de Melo, and G. Weikum, "WebChild 2.0 : Fine-grained commonsense knowledge distillation," in *Proceedings of ACL 2017, System Demonstrations*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 115–120. [Online]. Available: https://aclanthology.org/P17-4020 [Cited on page 19.]

[93] K. K. Schuler, "Verbnet: A broad-coverage, comprehensive verb lexicon," Ph.D. dissertation, University of Pennsylvania, 2006. [Online]. Available: http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf [Cited on page 19.]

[94] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. [Cited on page 19.]

[95] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, in *Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations*, 2016. [Online]. Available: https://arxiv.org/abs/1602.07332 [Cited on page 19.]

[96] P. Kapanipathi, V. Thost, S. Sankalp Patel, S. Whitehead, I. Abdelaziz, A. Balakrishnan, M. Chang, K. Fadnis, C. Gunasekara, B. Makni, N. Mattei, K. Talamadupula, and A. Fokoue, "Infusing knowledge into the textual entailment task using graph convolutional networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8074–8081, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6318 [Cited on page 19.]

[97] B. Y. Lin, X. Chen, J. Chen, and X. Ren, "KagNet: Knowledge-aware graph networks for commonsense reasoning," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2829–2839. [Online]. Available: https://aclanthology.org/D19-1282 [Cited on page 19.]

[98] P. Ammanabrolu, E. Tien, W. Cheung, Z. Luo, W. Ma, L. J. Martin, and M. O. Riedl, "Story realization: Expanding plot events into sentences," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 7375–7382, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6232 [Cited on page 19.]

[99] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1321–1330. [Online]. Available: https://proceedings.mlr.press/v70/guo17a.html [Cited on page 20.]

[100] E. Reif, A. Yuan, M. Wattenberg, F. B. Viegas, A. Coenen, A. Pearce, and B. Kim, "Visualizing and measuring the geometry of bert," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlche Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, pp. 8594–8603. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/159c1ffe5b61b41b3c4d8f4c2150f6c4-Paper.pdf [Cited on page 20.]

[101] G. Chronis and K. Erk, "When is a bishop not like a rook? when it's like a rabbi! multi-prototype BERT embeddings for estimating semantic relationships," in *Proceedings of the 24th Conference on Computational Natural Language Learning*. Online: Association for Computational Linguistics, Nov. 2020, pp. 227–244. [Online]. Available: https://aclanthology.org/2020.conll-1.17 [Cited on page 20.]

[102] I. Vulić, E. M. Ponti, R. Litschko, G. Glavaš, and A. Korhonen, "Probing pretrained language models for lexical semantics," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association

for Computational Linguistics, Nov. 2020, pp. 7222–7240. [Online]. Available: https://aclanthology.org/2020.emnlp-main.586 [Cited on page 20.]

[103] E. Voita, R. Sennrich, and I. Titov, "The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4396–4406. [Online]. Available: https://aclanthology.org/D19-1448 [Cited on page 20.]

[104] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. comput. appl. math. 20, 53-65," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 11 1987. [Cited on page 20.]

[105] T. Blevins and L. Zettlemoyer, "Moving down the long tail of word sense disambiguation with gloss informed bi-encoders," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 1006–1017. [Online]. Available: https://aclanthology.org/2020.acl-main.95 [Cited on page 24.]

[106] E. Barba, L. Procopio, and R. Navigli, "ConSeC: Word sense disambiguation as continuous sense comprehension," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1492–1503. [Online]. Available: https://aclanthology.org/2021.emnlp-main.112 [Cited on page 24.]

[107] M. Maru, S. Conia, M. Bevilacqua, and R. Navigli, "Nibbling at the hard core of Word Sense Disambiguation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4724–4737. [Online]. Available: https://aclanthology.org/2022.acl-long.324 [Cited on page 24.]

[108] Z. Wang, A. W. Yu, O. Firat, and Y. Cao, "Towards zero-label language learning," 2021. [Online]. Available: https://arxiv.org/abs/2109.09193 [Cited on page 24.]

[109] D. Colla, E. Mensa, and D. P. Radicioni, "Novel metrics for computing semantic similarity with sense embeddings," *Knowledge-Based Systems*, vol. 206, p. 106346, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120305025 [Cited on pages 24 and 26.]

[110] D. Fierens, G. Van den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt, "Inference and learning in probabilistic logic programs using weighted boolean formulas," *Theory and Practice of Logic Programming*, vol. 15, no. 3, pp. 358–401, 2015. [Cited on page 27.]

[111] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," in *Machine learning challenges workshop*. Springer, 2005, pp. 177–190. [Cited on page 27.]

[112] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html [Cited on page 27.]

[113] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. [Online]. Available: https://aclanthology.org/W18-5446 [Cited on page 27.]

[114] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf [Cited on page 27.]

[115] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "SWAG: A large-scale adversarial dataset for grounded commonsense inference," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association

for Computational Linguistics, Oct.-Nov. 2018, pp. 93–104. [Online]. Available: https://aclanthology.org/D18-1009 [Cited on page 27.]

[116] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela, "Adversarial NLI: A new benchmark for natural language understanding," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 4885–4901. [Online]. Available: https://aclanthology.org/2020.acl-main.441 [Cited on page 27.]

[117] D. Loureiro and A. Jorge, "Affordance extraction and inference based on semantic role labeling," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 91–96. [Online]. Available: https://aclanthology.org/W18-5514 [Cited on page 28.]

[118] S. Oliveira, D. Loureiro, and A. Jorge, "Improving portuguese semantic role labeling with transformers and transfer learning," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, 2021, pp. 1–9. [Cited on page 28.]

[119] J. Lajus, L. Galárraga, and F. Suchanek, "Fast and exact rule mining with amie 3," in *The Semantic Web*, A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase, and M. Cochez, Eds. Cham: Springer International Publishing, 2020, pp. 36–52. [Cited on page 28.]

[120] J. Stacey, P. Minervini, H. Dubossarsky, and M. Rei, "Logical reasoning with span predictions: Span-level logical atoms for interpretable and robust nli models," 2022. [Online]. Available: https://arxiv.org/abs/2205.11432 [Cited on page 28.]

[121] Z. Chen, Q. Gao, and L. S. Moss, "NeuralLog: Natural language inference with joint neural and logical reasoning," in *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*. Online: Association for Computational Linguistics, Aug. 2021, pp. 78–88. [Online]. Available: https://aclanthology.org/2021.starsem-1.7 [Cited on page 28.]

[122] A.-L. Kalouli, R. Crouch, and V. de Paiva, "Hy-NLI: a hybrid system for natural language inference," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee

on Computational Linguistics, Dec. 2020, pp. 5235–5249. [Online]. Available: https://aclanthology.org/2020.coling-main.459 [Cited on page 28.]

[123] L. Weber, P. Minervini, J. Münchmeyer, U. Leser, and T. Rocktäschel, "NLProlog: Reasoning with weak unification for question answering in natural language," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6151–6161. [Online]. Available: https://aclanthology.org/P19-1618 [Cited on page 28.]

[124] P. Minervini, M. Bosnjak, T. Rocktäschel, S. Riedel, and E. Grefenstette, "Differentiable reasoning on large knowledge bases and natural language," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 5182–5190. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5962 [Cited on page 28.]

[125] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," 2022. [Online]. Available: https://arxiv.org/abs/2201.11903 [Cited on page 28.]

[126] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," 2022. [Online]. Available: https://arxiv.org/abs/2205.11916 [Cited on page 28.]

[127] G. Marcus and E. Davis, *Rebooting AI: Building Artificial Intelligence We Can Trust*. USA: Pantheon Books, 2019. [Cited on page 29.]

[128] X. Zhang, A. Bosselut, M. Yasunaga, H. Ren, P. Liang, C. D. Manning, and J. Leskovec, "GreaseLM: Graph REASoning enhanced language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=41e9o6cQPj [Cited on page 30.]

[129] E. M. Bender and A. Koller, "Climbing towards NLU: On meaning, form, and understanding in the age of data," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 5185–5198. [Online]. Available: https://aclanthology.org/2020.acl-main.463 [Cited on page 30.]

[130] W. Merrill, Y. Goldberg, R. Schwartz, and N. A. Smith, "Provable Limitations of Acquiring Meaning from Ungrounded Form: What Will Future Language Models Understand?" *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1047–1060, 09 2021. [Online]. Available: https://doi.org/10.1162/tacl_a_00412 [Cited on page 30.]

[131] G. Marcus, "The next decade in AI: four steps towards robust artificial intelligence," *CoRR*, vol. abs/2002.06177, 2020. [Online]. Available: https://arxiv.org/abs/2002.06177 [Cited on page 30.]

[132] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *CoRR*, vol. abs/1901.07291, 2019. [Online]. Available: http://arxiv.org/abs/1901.07291 [Cited on page 139.]

Appendix A

Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation

Submitted: March 2019; Published: July 2019; CORE: A*.

Daniel Loureiro and Alípio Jorge. 2019. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, p. 5682–5691, Florence, Italy. Association for Computational Linguistics. Published PDF: http://dx.doi.org/10.18653/v1/P19-1569.

Relevant Contributions

- Presents state-of-the-art result for WSD, first achieved with BERT.
- Proposes sense embeddings with full-coverage of WordNet, using propagation.
- Demonstrates effectiveness of k-NN in the latent space of NLMs for WSD.
- Introduces the Uninformed Sense Matching (USM) task.
- Proposes using glosses to improve quality of sense embeddings.
- High impact: +100 citations according to Google Scholar.

Return to Table of Contents

Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation

Daniel Loureiro, Alípio Mário Jorge
LIAAD - INESC TEC
Faculty of Sciences - University of Porto, Portugal
dloureiro@fc.up.pt, amjorge@fc.up.pt

Abstract

Contextual embeddings represent a new generation of semantic representations learned from Neural Language Modelling (NLM) that addresses the issue of meaning conflation hampering traditional word embeddings. In this work, we show that contextual embeddings can be used to achieve unprecedented gains in Word Sense Disambiguation (WSD) tasks. Our approach focuses on creating sense-level embeddings with full-coverage of WordNet, and without recourse to explicit knowledge of sense distributions or task-specific modelling. As a result, a simple Nearest Neighbors (k-NN) method using our representations is able to consistently surpass the performance of previous systems using powerful neural sequencing models. We also analyse the robustness of our approach when ignoring part-of-speech and lemma features, requiring disambiguation against the full sense inventory, and revealing shortcomings to be improved. Finally, we explore applications of our sense embeddings for concept-level analyses of contextual embeddings and their respective NLMs.

1 Introduction

Word Sense Disambiguation (WSD) is a core task of Natural Language Processing (NLP) which consists in assigning the correct sense to a word in a given context, and has many potential applications (Navigli, 2009). Despite breakthroughs in distributed semantic representations (i.e. word embeddings), resolving lexical ambiguity has remained a long-standing challenge in the field. Systems using non-distributional features, such as It Makes Sense (IMS, Zhong and Ng, 2010), remain surprisingly competitive against neural sequence models trained end-to-end. A baseline that simply chooses the most frequent sense (MFS) has also proven to be notoriously difficult to surpass.

Several factors have contributed to this limited progress over the last decade, including lack of standardized evaluation, and restricted amounts of sense annotated corpora. Addressing the evaluation issue, Raganato et al. (2017a) has introduced a unified evaluation framework that has already been adopted by the latest works in WSD. Also, even though SemCor (Miller et al., 1994) still remains the largest manually annotated corpus, supervised methods have successfully used label propagation (Yuan et al., 2016), semantic networks (Vial et al., 2018) and glosses (Luo et al., 2018b) in combination with annotations to advance the state-of-the-art. Meanwhile, task-specific sequence modelling architectures based on BiLSTMs or Seq2Seq (Raganato et al., 2017b) haven't yet proven as advantageous for WSD.

Until recently, the best semantic representations at our disposal, such as word2vec (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017), were bound to word types (i.e. distinct tokens), converging information from different senses into the same representations (e.g. 'play song' and 'play tennis' share the same representation of 'play'). These word embeddings were learned from unsupervised Neural Language Modelling (NLM) trained on fixed-length contexts. However, by recasting the same word types across different sense-inducing contexts, these representations became insensitive to the different senses of polysemous words. Camacho-Collados and Pilehvar (2018) refer to this issue as the meaning conflation deficiency and explore it more thoroughly in their work.

Recent improvements to NLM have allowed for learning representations that are context-specific and detached from word types. While word embedding methods reduced NLMs to fixed representations after pretraining, this new generation of contextual embeddings employs the pretrained

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5682–5691
Florence, Italy, July 28 - August 2, 2019.

NLM to infer different representations induced by arbitrarily long contexts. Contextual embeddings have already had a major impact on the field, driving progress on numerous downstream tasks. This success has also motivated a number of iterations on embedding models in a short timespan, from context2vec (Melamud et al., 2016), to GPT (Radford et al., 2018), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019).

Being context-sensitive by design, contextual embeddings are particularly well-suited for WSD. In fact, Melamud et al. (2016) and Peters et al. (2018) produced contextual embeddings from the SemCor dataset and showed competitive results on Raganato et al. (2017a)'s WSD evaluation framework, with a surprisingly simple approach based on Nearest Neighbors (k-NN). These results were promising, but those works only produced sense embeddings for the small fraction of WordNet (Fellbaum, 1998) senses covered by SemCor, resorting to the MFS approach for a large number of instances. Lack of high coverage annotations is one of the most pressing issues for supervised WSD approaches (Le et al., 2018).

Our experiments show that the simple k-NN w/MFS approach using BERT embeddings suffices to surpass the performance of all previous systems. Most importantly, in this work we introduce a method for generating sense embeddings with full-coverage of WordNet, which further improves results (additional 1.9% F1) while forgoing MFS fallbacks. To better evaluate the fitness of our sense embeddings, we also analyse their performance without access to lemma or part-of-speech features typically used to restrict candidate senses. Representing sense embeddings in the same space as any contextual embeddings generated from the same pretrained NLM eases introspections of those NLMs, and enables token-level intrinsic evaluations based on k-NN WSD performance. We summarize our contributions[1] below:

- A method for creating sense embeddings for all senses in WordNet, allowing for WSD based on k-NN without MFS fallbacks.
- Major improvement over the state-of-the-art on cross-domain WSD tasks, while exploring the strengths and weaknesses of our method.
- Applications of our sense embeddings for concept-level analyses of NLMs.

[1]Code and data: github.com/danlou/lmms

2 Language Modelling Representations

Distributional semantic representations learned from Unsupervised Neural Language Modelling (NLM) are currently used for most NLP tasks. In this section we cover aspects of word and contextual embeddings, learned from from NLMs, that are particularly relevant for our work.

2.1 Static Word Embeddings

Word embeddings are distributional semantic representations usually learned from NLM under one of two possible objectives: predict context words given a target word (Skip-Gram), or the inverse (CBOW) (word2vec, Mikolov et al., 2013). In both cases, context corresponds to a fixed-length window sliding over tokenized text, with the target word at the center. These modelling objectives are enough to produce dense vector-based representations of words that are widely used as powerful initializations on neural modelling architectures for NLP. As we explained in the introduction, word embeddings are limited by meaning conflation around word types, and reduce NLM to fixed representations that are insensitive to contexts. However, with fastText (Bojanowski et al., 2017) we're not restricted to a finite set of representations and can compositionally derive representations for word types unseen during training.

2.2 Contextual Embeddings

The key differentiation of contextual embeddings is that they are context-sensitive, allowing the same word types to be represented differently according to the contexts in which they occurr. In order to be able to produce new representations induced by different contexts, contextual embeddings employ the pretrained NLM for inferences. Also, the NLM objective for contextual embeddings is usually directional, predicting the previous and/or next tokens in arbitrarily long contexts (usually sentences). ELMo (Peters et al., 2018) was the first implementation of contextual embeddings to gain wide adoption, but it was shortly after followed by BERT (Devlin et al., 2019) which achieved new state-of-art results on 11 NLP tasks. Interestingly, BERT's impressive results were obtained from task-specific fine-tuning of pretrained NLMs, instead of using them as features in more complex models, emphasizing the quality of these representations.

3 Word Sense Disambiguation (WSD)

There are several lines of research exploring different approaches for WSD (Navigli, 2009). Supervised methods have traditionally performed best, though this distinction is becoming increasingly blurred as works in supervised WSD start exploiting resources used by knowledge-based approaches (e.g. Luo et al., 2018a; Vial et al., 2018). We relate our work to the best-performing WSD methods, regardless of approach, as well as methods that may not perform as well but involve producing sense embeddings. In this section we introduce the components and related works that are most relevant for our approach.

3.1 Sense Inventory, Attributes and Relations

The most popular sense inventory is WordNet, a semantic network of general domain concepts linked by a few relations, such as synonymy and hypernymy. WordNet is organized at different abstraction levels, which we describe below. Following the notation used in related works, we represent the main structure of WordNet, called synset, with $lemma^{\#}_{POS}$, where $lemma$ corresponds to the canonical form of a word, $_{POS}$ corresponds to the sense's part-of-speech (noun, verb, adjective or adverb), and $^{\#}$ further specifies this entry.

- Synsets: groups of synonymous words that correspond to the same sense, e.g. dog^1_n.
- Lemmas: canonical forms of words, may belong to multiple synsets, e.g. *dog* is a lemma for dog^1_n and $chase^1_v$, among others.
- Senses: lemmas specifed by sense (i.e. sensekeys), e.g. *dog%1:05:00::*, and *domestic_dog%1:05:00::* are senses of dog^1_n.

Each synset has a number of attributes, of which the most relevant for this work are:

- Glosses: dictionary definitions, e.g. dog^1_n has the definition 'a member of the genus Ca...'.
- Hypernyms: 'type of' relations between synsets, e.g. dog^1_n is a hypernym of pug^1_n.
- Lexnames: syntactical and logical groupings, e.g. the lexname for dog^1_n is *noun.animal*.

In this work we're using WordNet 3.0, which contains 117,659 synsets, 206,949 unique senses, 147,306 lemmas, and 45 lexnames.

3.2 WSD State-of-the-Art

While non-distributional methods, such as Zhong and Ng (2010)'s IMS, still perform competitively, there are have been several noteworthy advancements in the last decade using distributional representations from NLMs. Iacobacci et al. (2016) improved on IMS's performance by introducing word embeddings as additional features.

Yuan et al. (2016) achieved significantly improved results by leveraging massive corpora to train a NLM based on an LSTM architecture. This work is contemporaneous with Melamud et al. (2016), and also uses a very similar approach for generating sense embeddings and relying on k-NN w/MFS for predictions. Although most performance gains stemmed from their powerful NLM, they also introduced a label propagation method that further improved results in some cases. Curiously, the objective Yuan et al. (2016) used for NLM (predicting held-out words) is very evocative of the cloze-style Masked Language Model introduced by Devlin et al. (2019). Le et al. (2018) replicated this work and offers additional insights.

Raganato et al. (2017b) trained neural sequencing models for end-to-end WSD. This work reframes WSD as a translation task where sequences of words are translated into sequences of senses. The best result was obtained with a BiLSTM trained with auxilliary losses specific to parts-of-speech and lexnames. Despite the sophisticated modelling architecture, it still performed on par with Iacobacci et al. (2016).

The works of Melamud et al. (2016) and Peters et al. (2018) using contextual embeddings for WSD showed the potential of these representations, but still performed comparably to IMS.

Addressing the issue of scarce annotations, recent works have proposed methods for using resources from knowledge-based approaches. Luo et al. (2018a) and Luo et al. (2018b) combine information from glosses present in WordNet, with NLMs based on BiLSTMs, through memory networks and co-attention mechanisms, respectively. Vial et al. (2018) follows Raganato et al. (2017b)'s BiLSTM method, but leverages the semantic network to strategically reduce the set of senses required for disambiguating words.

All of these works rely on MFS fallback. Additionally, to our knowledge, all also perform disambiguation only against the set of admissible senses given the word's lemma and part-of-speech.

3.3 Other methods with Sense Embeddings

Some works may no longer be competitive with the state-of-the-art, but nevertheless remain relevant for the development of sense embeddings. We recommend the recent survey of Camacho-Collados and Pilehvar (2018) for a thorough overview of this topic, and highlight a few of the most relevant methods. Chen et al. (2014) initializes sense embeddings using glosses and adapts the Skip-Gram objective of word2vec to learn and improve sense embeddings jointly with word embeddings. Rothe and Schütze (2015)'s AutoExtend method uses pretrained word2vec embeddings to compose sense embeddings from sets of synonymous words. Camacho-Collados et al. (2016) creates the NASARI sense embeddings using structural knowledge from large multilingual semantic networks.

These methods represent sense embeddings in the same space as the pretrained word embeddings, however, being based on fixed embedding spaces, they are much more limited in their ability to generate contextual representations to match against. Furthermore, none of these methods (or those in §3.2) achieve full-coverage of the +200K senses in WordNet.

4 Method

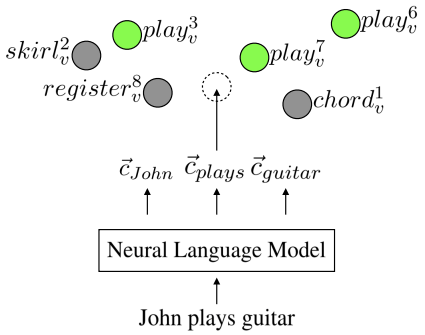


Figure 1: Illustration of our k-NN approach for WSD, which relies on full-coverage sense embeddings represented in the same space as contextualized embeddings. For simplification, we label senses as synsets. Grey nodes belong to different lemmas (see §5.3).

Our WSD approach is strictly based on k-NN (see Figure 1), unlike any of the works referred previously. We avoid relying on MFS for lemmas that do not occur in annotated corpora by generating sense embeddings with full-coverage of WordNet. Our method starts by generating sense embeddings from annotations, as done by other works, and then introduces several enhancements towards full-coverage, better performance and increased robustness. In this section, we cover each of these techniques.

4.1 Embeddings from Annotations

Our set of full-coverage sense embeddings is bootstrapped from sense-annotated corpora. Sentences containing sense-annotated tokens (or spans) are processed by a NLM in order to obtain contextual embeddings for those tokens. After collecting all sense-labeled contextual embeddings, each sense embedding is determined by averaging its corresponding contextual embeddings. Formally, given n contextual embeddings $\vec{c}$ for some sense s:

$$\vec{v}_s = \frac{1}{n}\sum_{i=1}^{n}\vec{c}_i, dim(\vec{v}_s) = 1024$$

In this work we use pretrained ELMo and BERT models to generate contextual embeddings. These models can be identified and replicated with the following details:

- ELMo: 1024 (2x512) embedding dimensions, 93.6M parameters. Embeddings from top layer (2).
- BERT: 1024 embedding dimensions, 340M parameters, cased. Embeddings from sum of top 4 layers ([-1,-4])[2].

BERT uses WordPiece tokenization that doesn't always map to token-level annotations (e.g. 'multiplication' becomes 'multi', '##plication'). We use the average of subtoken embeddings as the token-level embedding. Unless specified otherwise, our LMMS method uses BERT.

4.2 Extending Annotation Coverage

As many have emphasized before (Navigli, 2009; Camacho-Collados and Pilehvar, 2018; Le et al., 2018), the lack of sense annotations is a major limitation of supervised approaches for WSD. We address this issue by taking advantage of the semantic relations in WordNet to extend the annotated signal to other senses. Semantic networks are often explored by knowledge-based approaches, and some recent works in supervised approaches as well (Luo et al., 2018a; Vial et al., 2018). The

[2]This was the configuration that performed best out of the ones on Table 7 of Devlin et al. (2018).

guiding principle behind these approaches is that sense-level representations can be imputed (or improved) from other representations that are known to correspond to generalizations due to the network's taxonomical structure. Vial et al. (2018) leverages relations in WordNet to reduce the sense inventory to a minimal set of entries, making the task easier to model while maintaining the ability to distinguish senses. We take the inverse path of leveraging relations to produce representations for additional senses.

On §3.1 we covered synsets, hypernyms and lexnames, which correspond to increasingly abstract generalizations. Missing sense embeddings are imputed from the aggregation of sense embeddings at each of these abstraction levels. In order to get embeddings that are representative of higher-level abstractions, we simply average the embeddings of all lower-level constituents. Thus, a synset embedding corresponds to the average of all of its sense embeddings, a hypernym embedding corresponds to the average of all of its synset embeddings, and a lexname embedding corresponds to the average of a larger set of synset embeddings. All lower abstraction representations are created before next-level abstractions to ensure that higher abstractions make use of lower generalizations. More formally, given all missing senses in WordNet $\hat{s} \in W$, their synset-specific sense embeddings $S_{\hat{s}}$, hypernym-specific synset embeddings $H_{\hat{s}}$, and lexname-specific synset embeddings $L_{\hat{s}}$, the procedure has the following stages:

$$(1) \quad if |S_{\hat{s}}| > 0, \qquad \vec{v}_{\hat{s}} = \frac{1}{|S_{\hat{s}}|} \sum \vec{v}_s, \forall \vec{v}_s \in S_{\hat{s}}$$

$$(2) \quad if |H_{\hat{s}}| > 0, \quad \vec{v}_{\hat{s}} = \frac{1}{|H_{\hat{s}}|} \sum \vec{v}_{syn}, \forall \vec{v}_{syn} \in H_{\hat{s}}$$

$$(3) \quad if |L_{\hat{s}}| > 0, \quad \vec{v}_{\hat{s}} = \frac{1}{|L_{\hat{s}}|} \sum \vec{v}_{syn}, \forall \vec{v}_{syn} \in L_{\hat{s}}$$

In Table 1 we show how much coverage extends while improving both recall and precision.

		F1 / P / R (without MFS)	
Source	**Coverage**	**BERT**	**ELMo**
SemCor	16.11%	68.9 / 72.4 / 65.7	63.0 / 66.2 / 60.1
+ synset	26.97%	70.0 / 72.6 / 70.0	63.9 / 66.3 / 61.7
+ hypernym	74.70%	73.0 / 73.6 / 72.4	67.2 / 67.7 / 66.6
+ lexname	100%	73.8 / 73.8 / 73.8	68.1 / 68.1 / 68.1

Table 1: Coverage of WordNet when extending to increasingly abstract representations along with performance on the ALL test set of Raganato et al. (2017a).

4.3 Improving Senses using the Dictionary

There's a long tradition of using glosses for WSD, perhaps starting with the popular work of Lesk (1986), which has since been adapted to use distributional representations (Basile et al., 2014). As a sequence of words, the information contained in glosses can be easily represented in semantic spaces through approaches used for generating sentence embeddings. There are many methods for generating sentence embeddings, but it's been shown that a simple weighted average of word embeddings performs well (Arora et al., 2017).

Our contextual embeddings are produced from NLMs using attention mechanisms, assigning more importance to some tokens over others, so they already come 'pre-weighted' and we embed glosses simply as the average of all of their contextual embeddings (without preprocessing). We've also found that introducing synset lemmas alongside the words in the gloss helps induce better contextualized embeddings (specially when glosses are short). Finally, we make our dictionary embeddings ($\vec{v}_d$) sense-specific, rather than synset-specific, by repeating the lemma that's specific to the sense, alongside the synset's lemmas and gloss words. The result is a sense-level embedding, determined without annotations, that is represented in the same space as the sense embeddings we described in the previous section, and can be trivially combined through concatenation or average for improved performance (see Table 2).

Our empirical results show improved performance by concatenation, which we attribute to preserving complementary information from glosses. Both averaging and concatenating representations (previously L_2 normalized) also serves to smooth possible biases that may have been learned from the SemCor annotations. Note that while concatenation effectively doubles the size of our embeddings, this doesn't equal doubling the expressiveness of the distributional space, since they're two representations from the same NLM. This property also allows us to make predictions for contextual embeddings (from the same NLM) by simply repeating those embeddings twice, aligning contextual features against sense and dictionary features when computing cosine similarity. Thus, our sense embeddings become:

$$\vec{v}_s = \begin{bmatrix} ||\vec{v}_s||_2 \\ ||\vec{v}_d||_2 \end{bmatrix}, dim(\vec{v}_s) = 2048$$

Configurations	LMMS$_{1024}$			LMMS$_{2048}$			LMMS$_{2348}$
Embeddings							
Contextual (d=1024)	✗		✗	✗	✗		✗
Dictionary (d=1024)		✗	✗	✗		✗	✗
Static (d=300)					✗	✗	✗
Operation							
Average			✗				
Concatenation				✗	✗	✗	✗
Perf. (F1 on ALL)							
Lemma & POS	73.8	58.7	75.0	**75.4**	73.9	58.7	**75.4**
Token (Uninformed)	42.7	6.1	36.5	35.1	64.4	45.0	**66.0**

Table 2: Overview of the different performance of various setups regarding choice of embeddings and combination strategy. All results are for the 1-NN approach on the ALL test set of Raganato et al. (2017a). We also show results that ignore the lemma and part-of-speech features of the test sets to show that the inclusion of static embeddings makes the method significantly more robust to real-world scenarios where such gold features may not be available.

4.4 Morphological Robustness

WSD is expected to be performed only against the set of candidate senses that are specific to a target word's lemma. However, as we'll explain in §5.3, there are cases where it's undesirable to restrict the WSD process.

We leverage word embeddings specialized for morphological representations to make our sense embeddings more resilient to the absence of lemma features, achieving increased robustness. This addresses a problem arising from the susceptibility of contextual embeddings to become entirely detached from the morphology of their corresponding tokens, due to interactions with other tokens in the sentence.

We choose fastText (Bojanowski et al., 2017) embeddings (pretrained on CommonCrawl), which are biased towards morphology, and avoid Out-of-Vocabulary issues as explained in §2.1. We use fastText to generate static word embeddings for the lemmas ($\vec{v}_l$) corresponding to all senses, and concatenate these word embeddings to our previous embeddings. When making predictions, we also compute fastText embeddings for tokens, allowing for the same alignment explained in the previous section. This technique effectively makes sense embeddings of morphologically related lemmas more similar. Empirical results (see Table 2) show that introducing these static embeddings is crucial for achieving satisfactory performance when not filtering candidate senses.

Our final, most robust, sense embeddings are thus:

$$\vec{v}_s = \begin{bmatrix} ||\vec{v}_s||_2 \\ ||\vec{v}_d||_2 \\ ||\vec{v}_l||_2 \end{bmatrix}, dim(\vec{v}_s) = 2348$$

5 Experiments

Our experiments centered on evaluating our solution on Raganato et al. (2017a)'s set of cross-domain WSD tasks. In this section we compare our results to the current state-of-the-art, and provide results for our solution when disambiguating against the full set of possible senses in WordNet, revealing shortcomings to be improved.

5.1 All-Words Disambiguation

In Table 3 we show our results for all tasks of Raganato et al. (2017a)'s evaluation framework. We used the framework's scoring scripts to avoid any discrepancies in the scoring methodology. Note that the k-NN referred in Table 3 always refers to the closest neighbor, and relies on MFS fallbacks.

The first noteworthy result we obtained was that simply replicating Peters et al. (2018)'s method for WSD using BERT instead of ELMo, we were able to significantly, and consistently, surpass the performance of all previous works. When using our method (LMMS), performance still improves significantly over the previous impressive results (+1.9 F1 on ALL, +3.4 F1 on SemEval 2013). Interestingly, we found that our method using ELMo embeddings didn't outperform ELMo k-NN with MFS fallback, suggesting that it's necessary to achieve a minimum competence level of embeddings from sense annotations (and glosses) before the inferred sense embeddings become more useful than MFS.

In Figure 2 we show results when considering additional neighbors as valid predictions, together with a random baseline considering that some target words may have less senses than the number of accepted neighbors (always correct).

Model	Senseval2 (n=2,282)	Senseval3 (n=1,850)	SemEval2007 (n=455)	SemEval2013 (n=1,644)	SemEval2015 (n=1,022)	ALL (n=7,253)
MFS† (Most Frequent Sense)	65.6	66.0	54.5	63.8	67.1	64.8
IMS† (2010)	70.9	69.3	61.3	65.3	69.5	68.4
IMS + embeddings† (2016)	72.2	70.4	62.6	65.9	71.5	69.6
context2vec k-NN† (2016)	71.8	69.1	61.3	65.6	71.9	69.0
word2vec k-NN (2016)	67.8	62.1	58.5	66.1	66.7	-
LSTM-LP (Label Prop.) (2016)	73.8	71.8	63.5	69.5	72.6	-
Seq2Seq (Task Modelling) (2017b)	70.1	68.5	63.1*	66.5	69.2	68.6*
BiLSTM (Task Modelling) (2017b)	72.0	69.1	64.8*	66.9	71.5	69.9*
ELMo k-NN (2018)	71.5	67.5	57.1	65.3	69.9	67.9
HCAN (Hier. Co-Attention) (2018a)	72.8	70.3	-*	68.5	72.8	-*
BiLSTM w/Vocab. Reduction (2018)	72.6	70.4	61.5	70.8	71.3	70.8
BERT k-NN	**76.3**	73.2	66.2	71.7	74.1	73.5
LMMS_{2348} (ELMo)	68.1	64.7	53.8	66.9	69.0	66.2
LMMS_{2348} (BERT)	**76.3**	**75.6**	**68.1**	**75.1**	**77.0**	**75.4**

Table 3: Comparison with other works on the test sets of Raganato et al. (2017a). All works used sense annotations from SemCor as supervision, although often different pretrained embeddings. † - reproduced from Raganato et al. (2017a); * - used as a development set; bold - new state-of-the-art (SOTA); underlined - previous SOTA.

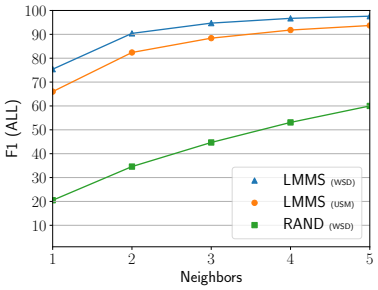


Figure 2: Performance gains with LMMS_{2348} when accepting additional neighbors as valid predictions.

5.2 Part-of-Speech Mismatches

The solution we introduced in §4.4 addressed missing lemmas, but we didn't propose a solution that addressed missing POS information. Indeed, the confusion matrix in Table 4 shows that a large number of target words corresponding to verbs are wrongly assigned senses that correspond to adjectives or nouns. We believe this result can help motivate the design of new NLM tasks that are more capable of distinguishing between verbs and non-verbs.

WN-POS	NOUN	VERB	ADJ	ADV
NOUN	96.95%	1.86%	0.86%	0.33%
VERB	9.08%	70.82%	19.98%	0.12%
ADJ	4.50%	0%	92.27%	2.93%
ADV	2.02%	0.29%	2.60%	95.09%

Table 4: POS Confusion Matrix for Uninformed Sense Matching on the ALL testset using LMMS_{2348}.

5.3 Uninformed Sense Matching

WSD tasks are usually accompanied by auxilliary parts-of-speech (POSs) and lemma features for restricting the number of possible senses to those that are specific to a given lemma and POS. Even if those features aren't provided (e.g. real-world applications), it's sensible to use lemmatizers or POS taggers to extract them for use in WSD. However, as is the case with using MFS fallbacks, this filtering step obscures the true impact of NLM representations on k-NN solutions.

Consequently, we introduce a variation on WSD, called Uninformed Sense Matching (USM), where disambiguation is always performed against the full set of sense embeddings (i.e. +200K vs. a maximum of 59). This change makes the task much harder (results on Table 2), but offers some insights into NLMs, which we cover briefly in §5.4.

5.4 Use of World Knowledge

It's well known that WSD relies on various types of knowledge, including commonsense and selectional preferences (Lenat et al., 1986; Resnik, 1997), for example. Using our sense embeddings for Uninformed Sense Matching allows us to glimpse into how NLMs may be interpreting contextual information with regards to the knowledge represented in WordNet. In Table 5 we show a few examples of senses matched at the token-level, suggesting that entities were topically understood and this information was useful to disambiguate verbs. These results would be less conclusive without full-coverage of WordNet.

Marlon⋆	**Brando**⋆	**played**	**Corleone**⋆	**in**	**Godfather**⋆
$person_n^1$	$person_n^1$	act_v^3	$syndicate_n^1$	$movie_n^1$	$location_n^1$
$womanizer_n^1$	$group_n^1$	$make_v^{42}$	$mafia_n^1$	$telefilm_n^1$	$here_n^1$
$bustle_n^1$	$location_n^1$	$emote_v^1$	$person_n^1$	$final\_cut_n^1$	$there_n^1$

$\mathbf{act_v^3}$: play a role or part; $\mathbf{make_v^{42}}$: represent fictiously, as in a play, or pretend to be or act like; $\mathbf{emote_v^1}$: give expression or emotion to, in a stage or movie role.

Serena⋆	**Williams**	**played**	**Kerber**⋆	**in**	**Wimbledon**⋆
$person_n^1$	$professional\_tennis_n^1$	$play_v^1$	$person_n^1$	win_v^1	$tournament_n^1$
$therefore_r^1$	$tennis_n^1$	$line\_up_v^6$	$group_n^1$	$romp_v^3$	$world\_cup_n^1$
$reef_n^1$	$singles_n^1$	$curl_v^5$	$take\_orders_v^2$	$carry_v^{38}$	$elimination\_tournament_n^1$

$\mathbf{play_v^1}$: participate in games or sport; $\mathbf{line\_up_v^6}$: take one's position before a kick-off; $\mathbf{curl_v^5}$: play the Scottish game of curling.

David	**Bowie**⋆	**played**	**Warszawa**⋆	**in**	**Tokyo**
$person_n^1$	$person_n^1$	$play_v^{14}$	$poland_n^1$	$originate\_in_n^1$	$tokyo_n^1$
$amati_n^2$	$folk\_song_n^1$	$play_v^6$	$location_n^1$	in_r^1	$japan_n^1$
$guarnerius_n^3$	$fado_n^1$	$riff_v^2$	$here_n^1$	$take\_the\_field_v^2$	$japanese_a^1$

$\mathbf{play_v^{14}}$: perform on a certain location; $\mathbf{play_v^6}$: replay (as a melody); $\mathbf{riff_v^2}$: play riffs.

Table 5: Examples controlled for syntactical changes to show how the correct sense for 'played' can be induced accordingly with the mentioned entities, suggesting that disambiguation is supported by world knowledge learned during LM pretraining. Words with ⋆ never occurred in SemCor. Senses shown correspond to the top 3 matches in LMMS_{1024} for each token's contextual embedding (uninformed). For clarification, below each set of matches are the WordNet definitions for the top disambiguated senses of 'played'.

6 Other Applications

Analyses of conventional word embeddings have revealed gender or stereotype biases (Bolukbasi et al., 2016; Caliskan et al., 2017) that may have unintended consequences in downstream applications. With contextual embeddings we don't have sets of concept-level representations for performing similar analyses. Word representations can naturally be derived from averaging their contextual embeddings occurring in corpora, but then we're back to the meaning conflation issue described earlier. We believe that our sense embeddings can be used as representations for more easily making such analyses of NLMs. In Figure 3 we provide an example that showcases meaningful differences in gender bias, including for lemmas shared by different senses (*doctor*: PhD vs. medic, and *counselor*: therapist vs. summer camp supervisor). The bias score for a given synset s was calculated as following:

$$bias(s) = sim(\vec{v}_{man_n^1}, \vec{v}_s) - sim(\vec{v}_{woman_n^1}, \vec{v}_s)$$

Besides concept-level analyses, these sense embeddings can also be useful in applications that don't rely on a particular inventory of senses. In Loureiro and Jorge (2019), we show how similarities between matched sense embeddings and contextual embeddings are used for training a classifier that determines whether a word that occurs in two different sentences shares the same meaning.

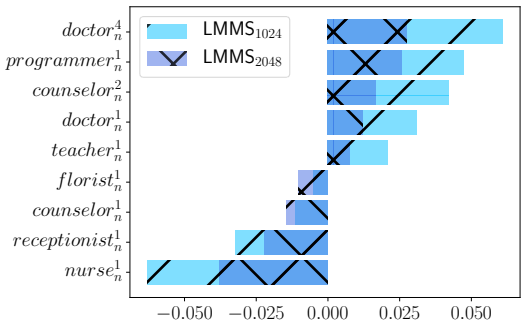


Figure 3: Examples of gender bias found in the sense vectors. Positive values quantify bias towards man_n^1, while negative values quantify bias towards $woman_n^1$.

7 Future Work

In future work we plan to use multilingual resources (i.e. embeddings and glosses) for improving our sense embeddings and evaluating on multilingual WSD. We're also considering exploring a semi-supervised approach where our best embeddings would be employed to automatically annotate corpora, and repeat the process described on this paper until convergence, iteratively fine-tuning sense embeddings. We expect our sense embeddings to be particularly useful in downstream tasks that may benefit from relational knowledge made accessible through linking words (or spans) to commonsense-level concepts in WordNet, such as Natural Language Inference.

8 Conclusion

This paper introduces a method for generating sense embeddings that allows a clear improvement of the current state-of-the-art on cross-domain WSD tasks. We leverage contextual embeddings, semantic networks and glosses to achieve full-coverage of all WordNet senses. Consequently, we're able to perform WSD with a simple 1-NN, without recourse to MFS fallbacks or task-specific modelling. Furthermore, we introduce a variant on WSD for matching contextual embeddings to all WordNet senses, offering a better understanding of the strengths and weaknesses of representations from NLM. Finally, we explore applications of our sense embeddings beyond WSD, such as gender bias analyses.

9 Acknowledgements

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project: UID/EEA/50014/2019.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR)*.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4356–4364, USA. Curran Associates Inc.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Int. Res.*, 63(1):743–788.

Jose Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36 – 64.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805v1.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Christiane Fellbaum. 1998. In *WordNet : an electronic lexical database*. MIT Press.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany. Association for Computational Linguistics.

Minh Le, Marten Postma, Jacopo Urbani, and Piek Vossen. 2018. A deep dive into word sense disambiguation with LSTM. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 354–365, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Doug Lenat, Mayank Prakash, and Mary Shepherd. 1986. Cyc: Using common sense knowledge to overcome brittleness and knowledge acquistion bottlenecks. *AI Mag.*, 6(4):65–85.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Daniel Loureiro and Alípio Mário Jorge. 2019. Liaad at semdeep-5 challenge: Word-in-context (wic). In *SemDeep-5@IJCAI 2019*, page forthcoming.

Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018a. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1402–1411, Brussels, Belgium. Association for Computational Linguistics.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018b. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482, Melbourne, Australia. Association for Computational Linguistics.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167, Copenhagen, Denmark. Association for Computational Linguistics.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In *Tagging Text with Lexical Semantics: Why, What, and How?*

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China. Association for Computational Linguistics.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2018. Improving the coverage and the generalization ability of neural word sense disambiguation through hypernymy and hyponymy relationships. *CoRR*, abs/1811.00960.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1374–1385, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.

Appendix B

LIAAD at SemDeep-5 Challenge: Word-in-Context (WiC)

Published: May 2019; Published: August 2019.

Daniel Loureiro and Alípio Jorge. 2019. In Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5), pages 1–5, Macau, China. Association for Computational Linguistics. Published PDF: https://aclanthology.org/W19-5801.pdf.

Relevant Contributions

- Demonstrates effectiveness of combining similarity between contextual embeddings and sense embeddings for a novel meaning change detection task.
- Reports competitive results without supervision from the task's training set, highlighting applicability of sense embeddings for other tasks besides WSD or USM.

Return to Table of Contents

LIAAD at SemDeep-5 Challenge: Word-in-Context (WiC)

Daniel Loureiro, Alípio Mário Jorge
LIAAD - INESC TEC
Faculty of Sciences - University of Porto, Portugal
dloureiro@fc.up.pt, amjorge@fc.up.pt

Abstract

This paper describes the LIAAD system that was ranked second place in the Word-in-Context challenge (WiC) featured in SemDeep-5. Our solution is based on a novel system for Word Sense Disambiguation (WSD) using contextual embeddings and full-inventory sense embeddings. We adapt this WSD system, in a straightforward manner, for the present task of detecting whether the same sense occurs in a pair of sentences. Additionally, we show that our solution is able to achieve competitive performance even without using the provided training or development sets, mitigating potential concerns related to task overfitting.

1 Task Overview

The Word-in-Context (WiC) (Pilehvar and Camacho-Collados, 2019) task aims to evaluate the ability of word embedding models to accurately represent context-sensitive words. In particular, it focuses on polysemous words which have been hard to represent as embeddings due to the meaning conflation deficiency (Camacho-Collados and Pilehvar, 2018). The task's objective is to detect if target words occurring in a pair of sentences carry the same meaning.

Recently, contextual word embeddings from ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019) have emerged as the successors to traditional embeddings. With this development, word embeddings have become context-sensitive by design and thus more suitable for representing polysemous words. However, as shown by the experiments of (Pilehvar and Camacho-Collados, 2019), they are still insufficient by themselves to reliably detect meaning shifts.

In this work, we propose a system designed for the larger task of Word Sense Disambiguation (WSD), where words are matched with specific senses, that can detect meaning shifts without being trained explicitly to do so. Our WSD system uses contextual word embeddings to produce sense embeddings, and has full-coverage of all senses present in WordNet 3.0 (Fellbaum, 1998). In Loureiro and Jorge (2019) we provide more details about this WSD system, called LMMS (Language Modelling Makes Sense), and demonstrate that it's currently state-of-the-art for WSD. For this challenge, we employ LMMS in two straightforward approaches: checking if the disambiguated senses are equal, and training a classifier based on the embedding similarities. Both approaches perform competitively, with the latter taking the second position in the challenge ranking, and the former trailing close behind even though it's tested directly on the challenge, forgoing the training and development sets.

2 System Description

LMMS has two useful properties: 1) uses contextual word embeddings to produce sense embeddings, and 2) covers a large set of over 117K senses from WordNet 3.0. The first property allows for comparing precomputed sense embeddings against contextual word embeddings generated at test-time (using the same language model). The second property makes the comparisons more meaningful by having a large selection of senses at disposal for comparison.

2.1 Sense Embeddings

Given the meaning conflation deficiency issue with traditional word embeddings, several works have focused on adapting Neural Language Models (NLMs) to produce word embeddings that are more sense-specific. In this work, we start producing sense embeddings from the approach used by recent works in contextual word embeddings, particularly context2vec (Melamud et al., 2016) and

ELMo (Peters et al., 2018), and introduce some improvements towards full-coverage and more accurate representations.

2.1.1 Using Supervision

Our set of full-coverage WordNet sense embeddings is bootstrapped from the SemCor corpus (Miller et al., 1994). Sentences containing sense-annotated tokens (or spans) are processed by a NLM in order to obtain contextual embeddings for those tokens. After collecting all sense-labeled contextual embeddings, each sense embedding ($\vec{v}_s$) is determined by averaging its corresponding contextual embeddings. Formally, given n contextual embeddings $\vec{c}$ for some sense s:

$$\vec{v}_s = \frac{1}{n}\sum_{i=1}^{n}\vec{c}_i$$

In this work, we used BERT as our NLM. For replicability, these are the relevant details: 1024 embedding dimensions, 340M parameters, cased. Embeddings result from the sum of top 4 layers ([-1, -4]). Moreover, since BERT uses WordPiece tokenization that doesn't always map to token-level annotations, we use the average of subtoken embeddings as the token-level embedding.

2.1.2 Extending Supervision

Despite its age, SemCor is still the largest sense-annotated corpus. The lack of larger sets of sense annotations is a major limitation of supervised approaches for WSD (Le et al., 2018). We address this issue by taking advantage of the semantic relations in WordNet to extend the annotated signal to other senses. Missing sense embeddings are inferred (i.e. imputed) from the aggregation of sense embeddings at different levels of abstraction from WordNet's ontology. Thus, a synset embedding corresponds to the average of all of its sense embeddings, a hypernym embedding corresponds to the average of all of its synset embeddings, and a lexname embedding corresponds to the average of a larger set of synset embeddings. All lower abstraction representations are created before next-level abstractions to ensure that higher abstractions make use of lower-level generalizations. More formally, given all missing senses in WordNet $\hat{s} \in W$, their synset-specific sense embeddings $S_{\hat{s}}$, hypernym-specific synset embeddings $H_{\hat{s}}$, and lexname-specific synset embeddings $L_{\hat{s}}$, the procedure has the following stages:

$$(1)\quad if|S_{\hat{s}}| > 0, \qquad \vec{v}_{\hat{s}} = \tfrac{1}{|S_{\hat{s}}|}\textstyle\sum \vec{v}_s, \forall \vec{v}_s \in S_{\hat{s}}$$

$$(2)\quad if|H_{\hat{s}}| > 0, \quad \vec{v}_{\hat{s}} = \tfrac{1}{|H_{\hat{s}}|}\textstyle\sum \vec{v}_{syn}, \forall \vec{v}_{syn} \in H_{\hat{s}}$$

$$(3)\quad if|L_{\hat{s}}| > 0, \quad \vec{v}_{\hat{s}} = \tfrac{1}{|L_{\hat{s}}|}\textstyle\sum \vec{v}_{syn}, \forall \vec{v}_{syn} \in L_{\hat{s}}$$

2.1.3 Leveraging Glosses

There's a long tradition of using glosses for WSD, perhaps starting with the popular work of Lesk (1986). As a sequence of words, the information contained in glosses can be easily represented in semantic spaces through approaches used for generating sentence embeddings. While there are many methods for generating sentence embeddings, it's been shown that a simple weighted average of word embeddings performs well (Arora et al., 2017).

Our contextual embeddings are produced from NLMs that employ attention mechanisms, assigning more importance to some tokens over others. As such, these embeddings already come 'pre-weighted' and we embed glosses simply as the average of all of their contextual embeddings (without preprocessing). We've found that introducing synset lemmas alongside the words in the gloss helps induce better contextualized embeddings (specially when glosses are short). Finally, we make our dictionary embeddings ($\vec{v}_d$) sense-specific, rather than synset-specific, by repeating the lemma that's specific to the sense alongside all of the synset's lemmas and gloss words. The result is a sense-level embedding that is represented in the same space as the embeddings we described in the previous section, and can be trivially combined through concatenation (previously L_2 normalized).

Given that both representations are based on the same NLM, we can make predictions for contextual embeddings of target words w (again, using the same NLM) at test-time by simply duplicating those embeddings, aligning contextual features against sense and dictionary features when computing cosine similarity. Thus, we have sense embeddings $\vec{v}_s$, to be matched against duplicated contextual embeddings $\vec{c}_w$, represented as follows:

$$\vec{v}_s = \begin{bmatrix} ||\vec{v}_s||_2 \\ ||\vec{v}_d||_2 \end{bmatrix}, \vec{c}_w = \begin{bmatrix} ||\vec{c}_w||_2 \\ ||\vec{c}_w||_2 \end{bmatrix}$$

2.2 Sense Disambiguation

Having produced our set of full-coverage sense embeddings, we perform WSD using a simple Nearest-Neighbors (k-NN) approach, similarly to Melamud et al. (2016) and Peters et al. (2018). We match the contextual word embedding of a target word against the sense embeddings that share the word's lemma (see Figure 1). Matching is performed using cosine similarity (with duplicated features on the contextual embedding for alignment, as explained in 2.1.3), and the top match is used as the disambiguated sense.

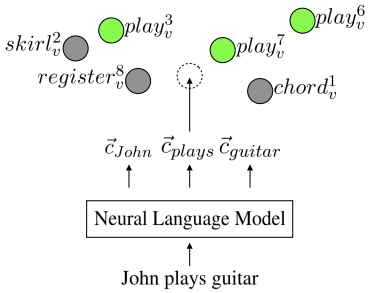


Figure 1: Illustration of our k-NN approach for WSD, which relies on full-coverage sense embeddings represented in the same space as contextualized embeddings.

2.3 Binary Classification

The WiC task calls for a binary judgement on whether the meaning of a target word occurring in a pair of sentences is the same or not. As such, our most immediate solution is to perform WSD and base our decision on the resulting senses. This approach performs competitively, but we've still found it worthwhile to use WiC's data to train a classifier based on the strengths of similarities between contextual and sense embeddings. In this section we explore the details of both approaches.

2.3.1 Sense Comparison

Our first approach is a straightforward comparison of the disambiguated senses assigned to the target word in each sentence. Considering the example in Figure 2, this approach simply requires checking if the sense $cook_v^2$ assigned to 'makes' in the first sentence equals the sense $produce_v^2$ assigned to the same word in the second sentence.

2.3.2 Classifying Similarities

The WSD procedure we describe in this paper represents sense embeddings in the same space as contextual word embeddings. Our second approach exploits this property by considering the similarities (including between different embedding types) that can be seen in Figure 2. In this approach, we take advantage of WiC's training set to learn a Logistic Regression Binary Classifier based on different sets of similarities. The choice of Logistic Regression is due to its explainability and lightweight training, besides competitive performance. We use sklearn's implementation (v0.20.1), with default parameters.

3 Results

The best system we submitted during the evaluation period of the challenge was a Logistic Regression classifier trained on two similarity features (sim_1 and sim_2, or contextual and sense-level). We obtained slightly better results with a classifier trained on all four similarities shown in Figure 2, but were unable to submit that system due to the limit of a maximum of three submissions during evaluation. Interestingly, the simple approach described in 2.3.1 achieved a competitive performance of 66.3 accuracy, without being trained or fine-tuned on WiC's data. Performance of best entries and baselines can be seen on Table 1.

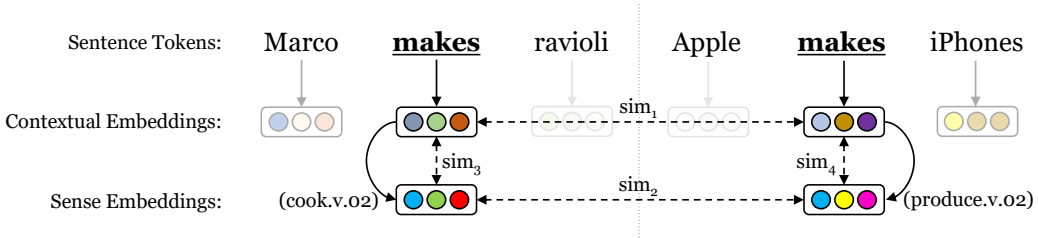


Figure 2: Components and interactions involved in our approaches. The sim_n labels correspond to cosine similarities between the related embeddings. Sense embeddings obtained from 1-NN matches of contextual embeddings.

Submission	Acc.
SuperGlue (Wang et al., 2019)	68.36
LMMS (Ours)	67.71
Ensemble (Soler et al., 2019)	66.71
ELMo-weighted (Ansell et al., 2019)	61.21
BERT-large	65.5
Context2vec	59.3
ELMo-3	56.5
Random	50.0

Table 1: Challenge results at the end of the evaluation period. Bottom results correspond to baselines.

4 Analysis

In this section we provide additional insights regarding our best approach. In Table 2, we show how task performance varies with the similarities considered.

Model	sim_n	Dev	Test
M0	N/A	68.18	66.29
M1	1	67.08	64.64
M2	2	66.93	66.21
M3	1, 2	68.50	67.71
M4	1, 2, 3, 4	69.12	**68.07**

Table 2: Accuracy of our different models. M0 wasn't trained on WiC data, the other models were trained on different sets of similarites. We submitted M3, but achieved slightly improved results with M4.

We determined that our best system (M4, using four features) obtains a precision of 0.65, recall of 0.82, and F1 of 0.73 on the development set, showing a relatively high proportion of false positives (21.6% vs. 9.25% of false negatives). This skewness can also be seen in the probability distribution chart at Figure 3. Additionally, we also present a ROC curve for this system at Figure 4 for a more detailed analysis of the system's performance.

5 Conclusion and Future Work

We've found that the WiC task can be adequately solved by systems trained for the larger task of WSD, specially if they're based on contextual embeddings, and when compared to the reported baselines. Still, we've found that the WiC dataset can be useful to learn a classifier that builds on top of the WSD system for improved performance on WiC's task of detecting shifts in meaning. In future work, we believe this improved ability to detect shifts in meaning can also assist WSD, particularly in generating semi-supervised datasets. We share our code and data at github.com/danlou/lmms.

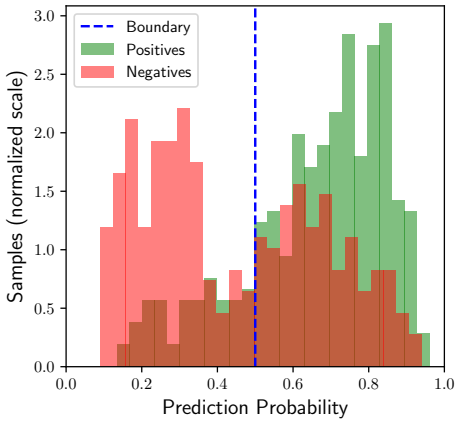


Figure 3: Distribution of Prediction Probabilities across labels, as evaluated by our best model on the development set.

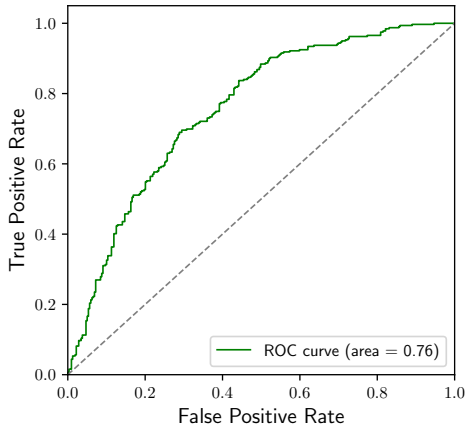


Figure 4: ROC curve for results of our best model on the development set.

Acknowledgements

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project: UID/EEA/50014/2019.

References

Alan Ansell, Felipe Bravo-Marquez, and Bernhard Pfahringer. 2019. An elmo-inspired approach to semdeep-5's word-in-context task. In *SemDeep-5@IJCAI 2019*, page forthcoming.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR)*.

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Int. Res.*, 63(1):743–788.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Christiane Fellbaum. 1998. In *WordNet : an electronic lexical database*. MIT Press.

Minh Le, Marten Postma, Jacopo Urbani, and Piek Vossen. 2018. A deep dive into word sense disambiguation with LSTM. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 354–365, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page forthcoming, Florence, Italy. Association for Computational Linguistics.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of NAACL*, Minneapolis, United States.

Aina Garí Soler, Marianna Apidianaki, and Alexandre Allauzen. 2019. Limsi-multisem at the ijcai semdeep-5 wic challenge: Context representations for word usage similarity estimation. In *SemDeep-5@IJCAI 2019*, page forthcoming.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *CoRR*, abs/1905.00537.

Appendix C

MedLinker: Medical Entity Linking with Neural Representations and Dictionary Matching

Submitted: October 2019; Published: April 2020; CORE: A.

Daniel Loureiro and Alípio Mário Jorge. 2020. In Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II. Springer-Verlag, Berlin, Heidelberg, 230–237. Published PDF: https://dl.acm.org/doi/10.1007/978-3-030-45442-5_29.

Social Media Adaptation: `https://tinyurl.com/MedLinkerSocial`

Relevant Contributions

- Proposes application of sense embedding methodology in the medical domain, using the UMLS ontology instead of WordNet.
- Combines Approximate Dictionary Matching with 1-NN for graceful degradation when it's not possible to represent majority of ontology as embeddings.

Return to Table of Contents

Appendix D

Don't Neglect the Obvious: On the Role of Unambiguous Words in Word Sense Disambiguation

Submitted: June 2020; Published: November 2020; CORE: A.

Daniel Loureiro and Jose Camacho-Collados. 2020. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3514–3520, Online. Association for Computational Linguistics. Published PDF: http://dx.doi.org/10.18653/v1/2020.emnlp-main.283.

Relevant Contributions

- Improves sense embeddings using automatic annotations of unambiguous words, increasing WordNet coverage by annotations from 16% to 57%.
- Adapts LMMS for RoBERTa (however, still using sum of last 4 layers).
- Presents T-SNE visualizations of sense embeddings: `https://danlou.github.io/uwa/`.

Return to Table of Contents

Don't Neglect the Obvious: On the Role of Unambiguous Words in Word Sense Disambiguation

Daniel Loureiro
LIAAD - INESC TEC
University of Porto, Portugal
dloureiro@fc.up.pt

Jose Camacho-Collados
School of Computer Science and Informatics
Cardiff University, United Kingdom
camachocolladosj@cardiff.ac.uk

Abstract

State-of-the-art methods for Word Sense Disambiguation (WSD) combine two different features: the power of pre-trained language models and a propagation method to extend the coverage of such models. This propagation is needed as current sense-annotated corpora lack coverage of many instances in the underlying sense inventory (usually WordNet). At the same time, unambiguous words make for a large portion of all words in WordNet, while being poorly covered in existing sense-annotated corpora. In this paper, we propose a simple method to provide annotations for most unambiguous words in a large corpus. We introduce the UWA (Unambiguous Word Annotations) dataset and show how a state-of-the-art propagation-based model can use it to extend the coverage and quality of its word sense embeddings by a significant margin, improving on its original results on WSD.

1 Introduction

There has been a lot of progress in word sense disambiguation (WSD) recently. This progress has been driven by two factors: (1) the introduction of large pre-trained Transformer-based language models and (2) propagation algorithms that extends the coverage of existing training sets. The gains due to pre-trained Neural Language Models (NLMs) such as BERT (Devlin et al., 2019) have been outstanding, helping reach levels close to human performance when training data is available. These models are generally based on a nearest neighbours strategy, where each sense is represented by a vector, exploiting the contextualized embeddings of these NLMs (Melamud et al., 2016; Peters et al., 2018; Loureiro and Jorge, 2019). However, training data for WSD is hard to obtain, and the most widely used training set nowadays, based on WordNet, dates back from the 90s (Miller et al., 1993, SemCor). This lack of curated data produces the so-called knowledge-acquisition bottleneck (Gale et al., 1992; Navigli, 2009).

However, there is a key source of information that has been neglected so far in existing sense-annotated corpora and propagation methods, which is the presence of unambiguous words from the underlying knowledge resource. Strikingly, WordNet, which is known to be a comprehensive resource, is mostly composed of unambiguous entries (30k lemmas are ambiguous, compared to 116k unambiguous). While the lack of unambiguous annotations does not have a direct effect in WSD, the fact that these unambiguous words are part of the same semantic network means they can have an effect on ambiguous words via standard propagation algorithms. These propagation algorithms start from a seed of senses occurring in the training data (and therefore their embeddings can be directly computed) and then propagate to the whole sense inventory via the semantic network (Vial et al., 2018; Loureiro and Jorge, 2019). Consequently, computing sense embeddings for unambiguous words can increase the number of seeds and improve the whole process. Covering these unambiguous words, however, is not an arduous task, as unlabelled corpora may suffice. We explore this hypothesis by labeling a large amount of unambiguous words in corpora extracted from the web, using WordNet as our reference sense inventory. While we can certainly find usages of a word not covered by WordNet, we found that our approach can obtain accurate occurrences with simple heuristics.

The contribution of this paper is twofold. First, we devise a simple methodology to construct UWA (Unambiguous Word Annotations), a large and, most importantly, diverse sense-annotated corpus that focuses on WordNet unambiguous words. Second, we show that by leveraging UWA, we can significantly improve a state-of-the-art WSD model.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 3514–3520,
November 16–20, 2020.

2 Related Work

The knowledge-acquisition bottleneck has been frequently addressed by automatically constructing sense-annotated corpora. Recent works propose methods that exploit knowledge from Wikipedia, such as NASARI vectors (Camacho-Collados et al., 2016), for providing sense annotations for concepts and entities (Scarlini et al., 2019; Pasini and Navigli, 2019). In the case of Scarlini et al. (2019), and similarly to Raganato et al. (2016), their method requires hyperlinks and category information from Wikipedia, hence not extensible to other kinds of corpora.[1] Previous approaches relied on parallel corpora for two or more languages. The OMSTI corpus (Taghipour and Ng, 2015) was constructed by exploiting the alignments of an English-Chinese corpus. Similarly, Delli Bovi et al. (2017) presented EuroSense, a multilingual sense-annotated corpus using the Europarl parallel corpus for 21 languages as reference. In contrast to these approaches, we focus on unambiguous senses and, therefore, are not constrained to only nouns, knowledge from Wikipedia, or a specific type of corpus.

Earlier works exploiting unambiguous words (Leacock et al., 1998; Mihalcea, 2002; Agirre and Martinez, 2004) and especially the subsequent extension by Martinez et al. (2008) are the most directly related to our paper. Martinez et al. (2008) retrieved example sentences with monosemous nouns from web search snippets and used them towards improved performance on WSD by leveraging WordNet relations. However, the WSD methods analyzed were sensitive to frequency bias, leading their collection effort to collect a large number of examples for fewer senses (and only nouns). In contrast, our solution is designed for all monosemous words, retrieving examples from web texts instead of snippets, attaining performance gains with even a single example per word.

3 Methodology

In this section we first explain our method to construct a corpus with unambiguous word annotations (Section 3.1). Then, we explain current models based on language models for WSD (Section 3.2) and describe a propagation method to infer additional OOV sense representations (Section 3.3).

[1] Pasini and Camacho-Collados (2020) provide a more detailed overview of existing sense-annotated corpora.

3.1 Unambiguous Word Annotations (UWA)

In order to properly test our hypothesis, we first require a sizable compilation of unambiguous words in context, particularly words that correspond to lemmas covered by WordNet. The extensiveness of WordNet means that most of its lemmas occur very rarely, and thus require processing large volumes of texts to achieve a high coverage. As such, in this work we develop the Unambiguous Word Annotations (UWA) corpus based on OpenWebText (Gokaslan and Cohen, 2019) and English Wikipedia (November 2019), processing over 53GB of texts from the web.

Each text is annotated for lemmas and part-of-speech using the Stanford CoreNLP toolkit (Manning et al., 2014). The annotations are filtered so that we only consider lemma/part-of-speech pairs that are present in WordNet, and correspond to a single sense (hence unambiguous), e.g., 'keypad/noun'. Naturally, some lemma/part-of-speech pairs may have additional meanings not covered in WordNet. For example, in "*Inception* was a box-office hit.", *Inception* makes reference to a movie and not to the unambiguous word *inception* from WordNet. To mitigate this issue, we applied Named Entity Recognition (NER) tagging, using spaCy (Honnibal and Montani, 2017), to discard lemmas that are recognized as entities but do not correspond to an entity in their WordNet sense. To this end, we leverage the entity annotations of WordNet synsets available in BabelNet (Navigli and Ponzetto, 2012). To keep the corpus at a reasonable size, we cut-off the maximum number of associated sentences (examples henceforth) per sense at 100.

Statistics. UWA covers a total of 98,494 senses, where 56.7% have 100 examples, and 81.2% have at least 10 examples. In Table 1 we show that UWA covers most senses for unambiguous words and, combined with SemCor, includes most senses in WordNet. This contrasts with other automatically-constructed datasets such as OMSTI (Taghipour and Ng, 2015) or T-o-M (Pasini and Navigli, 2019). These sense-annotated corpora, not aimed specifically at unambiguous words, have limited coverage in this respect, as they are mainly composed of annotations for senses already available in SemCor.

3.2 Neural Language Models for WSD

Recent NLMs, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), have been used with a high degree of success on WSD. They have

Corpus	# Instances Amb	Unamb	Avg # Exs	Coverage (w/ SC) Amb	Unamb	Total
SemCor	198,153	27,883	6.8	26.2	7.4	16.1
OMSTI	909,830	1,304	244.7	26.8	7.4	16.4
T-o-M	719,888	114,580	152.4	28.5	7.5	17.2
UWA(1)	0	98,494	1.0	26.2	82.9	56.7
UWA(10)	0	804,861	8.8	26.2	82.9	56.7
UWA(all)	0	6,111,453	54.1	26.2	82.9	56.7

Table 1: Number of instances, average number of examples per word sense, and coverage percentage (including SemCor) of various sense-annotated corpora.

been used differently depending on the nature of the disambiguation task: as feature providers for other neural architectures (Vial et al., 2019), simple classifiers after fine-tuning (Wang et al., 2019), or as generators of contextual embeddings to be matched through nearest neighbours (Melamud et al., 2016; Peters et al., 2018; Loureiro and Jorge, 2019; Reif et al., 2019, 1NN). Our experiments in this paper will focus on improving the latter type of approach. In particular, we will investigate the state-of-the-art LMMS model (Loureiro and Jorge, 2019). This model learns sense embeddings based on BERT states. These embeddings are then propagated through WordNet's ontology to infer additional senses, effectively providing a full coverage. While Loureiro and Jorge (2019) proposed variants of LMMS that combined propagation with gloss embeddings, or static embeddings, this paper is only concerned with the propagation method.

In our case, we essentially follow LMMS's layer pooling method to generate contextual embeddings for each sense occurrence in context (from a training set), and derive sense embeddings from the average of all corresponding contextual embeddings.

3.3 Network Propagation for Full-Coverage

The propagation method used in LMMS exploits the WordNet ontology to obtain a full coverage of sense embeddings from an initial set of embeddings based on a manually sense-annotated corpus like SemCor. This method explores different abstraction levels represented in WordNet: sets of synonyms (synsets), Is-A relations (hypernyms) and categorical groupings (lexnames[2]).

Initial sense embeddings are first used to compute synset embeddings as the average of all corresponding senses (analogously to how sense embeddings are computed from contextual embeddings). From that point, missing senses are represented by their corresponding synset embeddings. The remaining unrepresented senses are inferred from their hypernym and lexname embeddings, computed by averaging their neighbour synset embeddings. Note that this propagation process does not follow transitive relations in WordNet, i.e., a single synset's hypernym is considered, while the subsequent hypernyms along the root paths are ignored.

Since lexname embeddings can always be computed, this process can reach a full-coverage of WordNet starting with just the initial set of embeddings produced using SemCor. However, the set of SemCor embeddings only covers 16.1% of WordNet, so many of the inferred representations are redundant and therefore not entirely meaningful.

4 Evaluation

For our experiments we are interested in verifying the impact of using UWA to improve WSD performance. In particular, we test the unambiguous annotations of UWA as a complement of existing sense-annotated training data. To this end, as explained in Section 3, we make use of the state-of-the-art WSD model LMMS (Loureiro and Jorge, 2019). In addition to the original version using BERT, we also provide results with RoBERTa (Liu et al., 2019) for completeness. We use the 24-layer models for both BERT and RoBERTa.[3]

4.1 Word Sense Disambiguation (WSD)

Table 2 shows the WSD results on the standard evaluation framework of Raganato et al. (2017) for LMMS trained on the concatenation of SemCor and automatically-constructed corpora. In the table we include UWA with two different maximum number of examples per unambiguous word, i.e., 1 and 10. For comparison, we also include the results of EWISE (Kumar et al., 2019) and GlossBERT (Huang et al., 2019), which attempt to overcome the limited coverage of SemCor by exploiting textual definitions. As can be observed, the concatenation of our UWA corpus and SemCor provides the best overall results, regardless of the number of examples cut-off. Perhaps surprisingly, our corpus is the only one that provides improvements over the baseline (SemCor-only). These improvements are statistically significant on the full test set (i.e. ALL) for both BERT and RoBERTa with $p < 0.0005$, based on a t-test with respect to the

[2]Lexnames are also known as supersenses in the literature (Flekova and Gurevych, 2016; Pilehvar et al., 2017).

[3]Commonly referred to as *large* models.

	Corpus	SE-2	SE-3	SE07	SE13	SE15	ALL
LMMS-BERT	SC-noProp.	70.2	71.1	64.7	65.5	70.2	69.0
	SC-only	75.5	74.2	**66.8**	72.9	75.3	74.0
	OMSTI	73.7	68.8	63.5	**73.2**	74.8	71.9
	T-o-M	69.9	66.1	62.4	64.8	74.2	67.9
	UWA (1)	77.0	**74.2**	66.2	73.1	75.4	**74.5**
	UWA (10)	**77.3**	74.1	66.2	72.7	**75.7**	**74.5**
LMMS-RoBERTa	SC-noProp.	70.7	70.6	66.7	65.1	70.5	69.2
	SC-only	76.0	73.6	**69.2**	72.3	**75.9**	74.1
	OMSTI	73.4	70.1	66.6	71.5	74.6	71.9
	T-o-M	70.3	65.9	64.8	65.8	74.0	68.4
	UWA (1)	**77.8**	73.6	68.8	72.0	75.3	74.5
	UWA (10)	77.6	**73.7**	68.8	**72.7**	75.3	**74.6**
SOTA	$SC^{\ddagger}_{LMMS+}$	76.3	75.6	68.1	75.1	77.0	75.4
	$SC^{\dagger}_{Vial\ et\ al.}$	76.6	76.9	69.0	73.8	75.4	75.4
	$SC^{\ddagger\dagger}_{EWISE}$	73.8	71.1	67.3*	69.4	74.5	71.8
	$SC^{\ddagger\dagger}_{GlossBERT}$	77.7	75.2	72.5*	76.1	80.4	77.0

Table 2: F1 performance on the unified WSD evaluation framework. All corpora marked are concatenated with SemCor (SC). SOTAs reported for reference but not directly comparable due to use of definitions (‡) or not using a 1NN approach (†). All reported SOTAs are based on BERT trained on SC. Results in datasets that were used as development are marked with *.

Corpus	BERT			RoBERTa		
	F1	P@5	MRR	F1	P@5	MRR
OMSTI	50.2	66.0	57.5	44.1	59.9	51.7
T-o-M	45.8	62.1	53.3	42.1	60.7	50.2
UWA (10)	**54.9**	**74.1**	**63.5**	**62.1**	**80.2**	**70.1**

Table 3: Performance comparison in the uninformed setting. Each corpus is concatenated with SemCor.

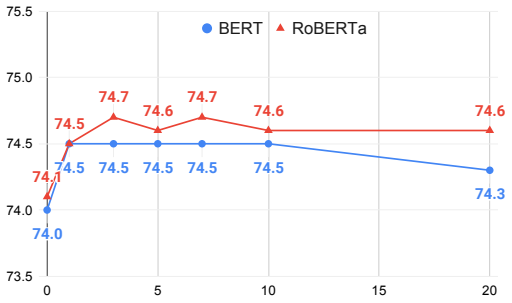


Figure 1: WSD performance (F1 on the ALL test set) with different numbers of UWA examples.

accuracy scores (equal to F1 in this setting). This can be explained by the fact that our corpus is the only one that significantly extends the coverage of SemCor, as explained in Section 3.1.

4.2 Uninformed Sense Matching (USM)

In standard WSD benchmarks, models are given the advantage of knowing the pre-defined set of possible senses before-hand. This is because gold PoS tags and lemmas are provided in these datasets. However, to better understand how robust a 1NN WSD model is, we can test it in an uninformed setting, i.e., where PoS tags and lemmas are not given and the model does not have access to the list of candidate senses. Instead, the model has to match senses from the whole sense inventory, unconstrained. Therefore, in this Uninformed Sense Matching (USM) setting we can use information retrieval ranking metrics with the model predictions (i.e. MRR or P@K) in addition to the standard F1. In line with the WSD results, Table 3 shows that UWA also substantially improves performance in the USM setting when comparing against currently available alternatives.

5 Analysis

In this section, we provide an analysis based on the number of examples (Section 5.1) and a visualization of the embedding space (Section 5.2).

5.1 Number of Examples

When compiling examples for learning sense representations, a natural question that arises is: how many examples are required to learn effective representations? The answer to this question can not only guide collection efforts, but also help clarify the requirements for learning effective representations in the simplest setting. To that end, we analyse the impact of using different number of examples from UWA on LMMS's WSD and USM performance. In Figure 1, we show the WSD performance trend using different number of examples per sense. As can be seen, performance improves substantially with only one example, and then stops improving after just two examples.

Similarly to our findings for WSD, Table 4 shows that a low number of examples, such as 2, already achieves the best overall results in the USM setting for BERT. Likewise, RoBERTa does not benefit from more than 5 examples. More generally, in USM the differences with respect to SemCor are more marked in comparison to the regular WSD setting. This is expected as the propagation algorithm has a stronger effect in this setting where all sense embeddings are considered.

5.2 Visualization of the Embedding Space

The propagation method used in LMMS is designed to backoff to increasingly abstract repre-

	BERT			RoBERTa		
Corpus	**F1**	**P@5**	**MRR**	**F1**	**P@5**	**MRR**
SemCor	52.5	67.1	59.2	58.0	72.8	64.7
UWA (1)	55.1	74.1	63.5	61.3	79.8	69.5
UWA (2)	**55.5**	**74.6**	**64.0**	61.8	**80.3**	70.0
UWA (3)	55.4	74.5	63.9	61.9	**80.3**	70.0
UWA (5)	55.4	74.4	63.8	**62.1**	**80.3**	**70.1**
UWA (7)	55.2	74.1	63.7	61.9	**80.3**	70.0
UWA (10)	54.9	74.1	63.5	**62.1**	80.2	**70.1**
UWA (20)	54.9	73.7	63.3	**62.1**	79.9	70.0

Table 4: USM performance of the LMMS model using SemCor and UWA with different example thresholds. Models tested on the concatenation of all WSD datasets of Raganato et al. (2017). As before, UWA is concatenated with SC in this experiment.

sentation levels, from synsets, to hypernyms, to supersenses (see Section 3.3 of the main paper). This naturally leads to a clustering effect, where many senses are represented with very similar, or equal, embeddings. In fact, we find that only 22% of sense embeddings learned from SemCor, and propagated following LMMS, are actually unique (remaining are shared by two or more senses). The addition of UWA increases this percentage to 68%.

To better understand this clustering effect, we used T-SNE (Maaten and Hinton, 2008) to visualize the WordNet synset embedding space. In Figure 2 we show synset embeddings learned from the SemCor+UWA(10) dataset, and learned from SemCor alone, both based on RoBERTa. While the same number of synset embeddings are learned in both cases, SemCor+UWA embeddings are better distributed across the vector space. This, in turn, causes a substantial reduction of high-density clusters, which stand in opposition to a rich distributional representation of senses.[4]

6 Conclusion

Unambiguous words are a surprisingly large portion of existing knowledge resources like WordNet. At the same time, their coverage in existing sense-annotated corpora is very limited. In this paper, we proposed a simple method which exploits sense annotations of unambiguous words from unlabeled corpora, thereby effectively extending existing sense-annotated corpora with low-effort. By leveraging a state-of-the-art BERT-based WSD system that propagates sense embeddings across WordNet, we have shown that these unambiguous words provide an excellent bridge to reach a wider range of OOV senses. This translates, in turn, into improving results for WSD. For future work it would be interesting to test these sense embeddings in a wider range of applications outside WSD. Since the embedding space is clearly more diversified, as shown in Figure 2, this may lead to improvements in other downstream tasks.

Moreover, one of the most surprising findings from this paper is that a single occurrence of OOV unambiguous words is enough to improve the performance of WSD models. This is relevant because (1) it is not always easy to retrieve a large number of examples for unambiguous words, and (2) it facilitates a cheaper manual verification, if required.

Finally, we openly release UWA, a large corpus annotated with unambiguous words, together improved BERT and RoBERTa-based sense embeddings, model predictions and visualizations at `http://danlou.github.io/uwa`.

Figure 2: T-SNE comparison of synset embeddings for whole WordNet learned from SC+UWA10 (top), or just SC (bottom). Colors represent source of annotations for embeddings (● SC ● UWA ● Propagation).

[4] We share interactive visualizations focusing on each of the 45 supersense groups (e.g. noun.communication) from WordNet at our UWA release website.

References

Eneko Agirre and David Martinez. 2004. Unsupervised WSD based on automatically retrieved examples: The importance of bias. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Barcelona, Spain. Association for Computational Linguistics.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64.

Claudio Delli Bovi, Jose Camacho-Collados, Alessandro Raganato, and Roberto Navigli. 2017. EuroSense: Automatic harvesting of multilingual sense annotations from parallel text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 594–600, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lucie Flekova and Iryna Gurevych. 2016. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2029–2041, Berlin, Germany. Association for Computational Linguistics.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5):415–439.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3507–3512, Hong Kong, China. Association for Computational Linguistics.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy. Association for Computational Linguistics.

Claudia Leacock, Martin Chodorow, and George A. Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

D. Martinez, O. Lopez de Lacalle, and E. Agirre. 2008. On the Use of Automatically Acquired Examples for All-Nouns Word Sense Disambiguation. *Journal of Artificial Intelligence Research*, 33:79–107.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Rada F. Mihalcea. 2002. Bootstrapping large sense tagged corpora. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, N.J.

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Tommaso Pasini and Jose Camacho-Collados. 2020. A short survey on sense-annotated corpora. In *Proceedings of the International Conference on Language Resources and Evaluation*, Marseille, France.

Tommaso Pasini and Roberto Navigli. 2019. Train-o-matic: Supervised word sense disambiguation with no (manual) effort. *Artificial Intelligence*, 279:103215.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a seamless integration of word senses into downstream NLP applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1857–1869, Vancouver, Canada. Association for Computational Linguistics.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2016. Automatic Construction and Evaluation of a Large Semantically Enriched Wikipedia. In *Proceedings of IJCAI*, pages 2894–2900, New York City, NY, USA.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. In *Advances in Neural Information Processing Systems*, pages 8592–8600.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2019. Just "OneSeC" for producing multilingual sense-annotated data. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 699–709, Florence, Italy. Association for Computational Linguistics.

Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344, Beijing, China. Association for Computational Linguistics.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2018. Improving the coverage and the generalization ability of neural word sense disambiguation through hypernymy and hyponymy relationships. *arXiv preprint arXiv:1811.00960*.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. In *Proceedings of the 10th Global WordNet Conference*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Appendix E

Analysis and Evaluation of Language Models for Word Sense Disambiguation

Submitted: August 2020; Published: July 2021; SJR: Q1.

Daniel Loureiro[†], Kiamehr Rezaee[†], Mohammad Taher Pilehvar, Jose Camacho-Collados († equal contribution). Computational Linguistics 2021; 47 (2): 387–443. Published PDF: https://doi.org/10.1162/COLI_a_00405.

Relevant Contributions

- Introduces the CoarseWSD-20 dataset for more thorough WSD evaluation (e.g., balanced, *n*-shots, out-of-domain, etc.)
- Presents alternative methods for WSD analysis (i.e., sense bias formula, cluster visualization, and cosine distribution plots)
- 1-NN outperforms fine-tuning in few-shot settings, with reduced frequency biases.

Return to Table of Contents

Analysis and Evaluation of Language Models for Word Sense Disambiguation

Daniel Loureiro*
LIAAD - INESC TEC
Department of Computer Science - FCUP
University of Porto
dloureiro@fc.up.pt

Kiamehr Rezaee*
Department of Computer Engineering
Iran University of Science and Technology
k_rezaee@comp.iust.ac.ir

Mohammad Taher Pilehvar
Tehran Institute for Advanced Studies
mp792@cam.ac.uk

Jose Camacho-Collados
School of Computer Science and Informatics
Cardiff University
camachocolladosj@cardiff.ac.uk

Transformer-based language models have taken many fields in NLP by storm. BERT and its derivatives dominate most of the existing evaluation benchmarks, including those for Word Sense Disambiguation (WSD), thanks to their ability in capturing context-sensitive semantic nuances. However, there is still little knowledge about their capabilities and potential limitations in encoding and recovering word senses. In this article, we provide an in-depth quantitative and qualitative analysis of the celebrated BERT model with respect to lexical ambiguity. One of the main conclusions of our analysis is that BERT can accurately capture high-level sense distinctions, even when a limited number of examples is available for each word sense. Our analysis also reveals that in some cases language models come close to solving coarse-grained noun disambiguation under ideal conditions in terms of availability of training data and computing resources. However, this scenario rarely occurs in real-world settings and, hence, many practical

*These authors contributed equally to this work.

Submission received: 19 August 2020; revised version received: 15 February 2021; accepted for publication: 4 March 2021.

https://doi.org/10.1162/COLI_a_00405

Computational Linguistics Volume 47, Number 2

challenges remain even in the coarse-grained setting. We also perform an in-depth comparison of the two main language model-based WSD strategies, namely, fine-tuning and feature extraction, finding that the latter approach is more robust with respect to sense bias and it can better exploit limited available training data. In fact, the simple feature extraction strategy of averaging contextualized embeddings proves robust even using only three training sentences per word sense, with minimal improvements obtained by increasing the size of this training data.

1. Introduction

In the past decade, word embeddings have undoubtedly been one of the major points of attention in research on lexical semantics. The introduction of Word2vec (Mikolov et al. 2013b), as one of the pioneering word *embedding* models, generated a massive wave in the field of lexical semantics, the impact of which is still being felt today. However, static word embeddings (such as Word2vec) suffer from the limitation of being *fixed* or context insensitive, that is, the word is associated with the same representation in all contexts, disregarding the fact that different contexts can trigger various meanings of the word, which might be even semantically unrelated. Sense representations were an attempt at addressing the meaning conflation deficiency of word embeddings (Reisinger and Mooney 2010; Camacho-Collados and Pilehvar 2018). Despite computing distinct representations for different senses of a word, hence addressing this deficiency of word embeddings, sense representations are not directly integrable into downstream NLP models. The integration usually requires additional steps, including a (non-optimal) disambiguation of the input text, which make sense embeddings fall short of fully addressing the problem.

The more recent *contextualized* embeddings (Peters et al. 2018a; Devlin et al. 2019) are able to simultaneously address both these limitations. Trained with language modeling objectives, contextualized models can compute *dynamic* meaning representations for words in context that highly correlate with humans' word sense knowledge (Nair, Srinivasan, and Meylan 2020). Moreover, contextualized embeddings provide a seamless integration into various NLP models, with minimal changes involved. Even better, given the extent of semantic and syntactic knowledge they capture, contextualized models get close to the one system for all tasks settings. Surprisingly, fine-tuning the same model on various target tasks often results in comparable or even higher performance when compared with sophisticated state-of-the-art task-specific models (Peters, Ruder, and Smith 2019). This has been shown for a wide range of NLP applications and tasks, including Word Sense Disambiguation (WSD), for which they have provided a significant performance boost, especially after the introduction of Transformer-based language models like BERT (Loureiro and Jorge 2019a; Vial, Lecouteux, and Schwab 2019; Wiedemann et al. 2019).

Despite their massive success, there has been limited work on the analysis of recent language models and on explaining the reasons behind their effectiveness in lexical semantics. Most analytical studies focus on syntax (Hewitt and Manning 2019; Saphra and Lopez 2019) or explore the behavior of self-attention heads (Clark et al. 2019) or layers (Tenney, Das, and Pavlick 2019), but there has been little work on investigating the potential of language models and their limitations in capturing other linguistic aspects, such as lexical ambiguity. Moreover, the currently popular language understanding evaluation benchmarks—for example, GLUE (Wang et al. 2018) and SuperGLUE (Wang et al. 2019)—mostly involve sentence-level representation, which does not shed much

light on the semantic properties of these models for individual words.[1] To our knowledge, there has so far been no in-depth analysis of the abilities of contextualized models in capturing the ambiguity property of words.

In this article, we carry out a comprehensive analysis to investigate how pretrained language models capture lexical ambiguity in the English language. Specifically, we scrutinize the two major language model-based WSD strategies (i.e., feature extraction and fine-tuning) under various disambiguation scenarios and experimental configurations. The main contributions of this work can be summarized as follows: (1) we provide an extensive quantitative evaluation of pretrained language models in standard WSD benchmarks; (2) we develop a new data set, CoarseWSD-20, which is particularly suited for the qualitative analysis of WSD systems; and (3) with the help of this data set, we perform an in-depth qualitative analysis and test the limitations of BERT on coarse-grained WSD. Data and code to reproduce all our experiments is available at `https://github.com/danlou/bert-disambiguation`.

The remainder of the article is organized as follows. In Section 2, we delineate the literature on probing pretrained language models and on analyzing the potential of representation models in capturing lexical ambiguity. We also describe in the same section the existing benchmarks for evaluating WSD. Section 3 presents an overview of WSD and its conventional paradigms. We then describe in the same section the two major approaches to utilizing language models for WSD, namely, nearest-neighbor feature extraction and fine-tuning. We also provide a quantitative comparison of some of the most prominent WSD approaches in each paradigm in various disambiguation scenarios, including fine- and coarse-grained settings. This quantitative analysis is followed by an analysis of models' performance per word categories (parts of speech) and for various layer-wise representations (in the case of language model-based techniques). Section 4 introduces CoarseWSD-20, the WSD data set we have constructed to facilitate our in-depth qualitative analysis. In Section 5 we evaluate the two major BERT-based WSD strategies on the benchmark. To highlight the improvement attributable to contextualized embeddings, we also provide results of a linear classifier based on pretrained FastText static word embeddings. Based on these experiments, we carry out an analysis on the impact of fine-tuning and also compare the two strategies with respect to robustness across domains and bias toward the most frequent sense. Section 6 reports our observations upon further scrutinizing the two strategies on a wide variety of settings such as few-shot learning and different training distributions. Section 7 summarizes the main results from the previous sections and discusses the main takeaways. Finally, Section 8 presents the concluding remarks and potential areas for future work.

2. Related Work

Recently, there have been several attempts at analyzing pretrained language models. In Section 2.1 we provide a general overview of the relevant works, and Section 2.2 covers those related to lexical ambiguity. Finally, in Section 2.3 we outline existing evaluation benchmarks for WSD, including CoarseWSD-20, which is the disambiguation data set we have constructed for our qualitative analysis.

1 WiC (Pilehvar and Camacho-Collados 2019) is the only SuperGLUE task where systems need to model the semantics of words in context (extended to several more languages in XL-WiC [Raganato et al. 2020]). In the Appendix we provide results for this task.

Computational Linguistics Volume 47, Number 2

2.1 Analysis of Pretrained Language Models

Despite their young age, pretrained language models, in particular, those based on Transformers, have now dominated the evaluation benchmarks for most NLP tasks (Devlin et al. 2019; Liu et al. 2019b). However, there has been limited work on understanding behind the scenes of these models.

Various studies have shown that fulfilling the language modeling objective inherently forces the model to capture various linguistic phenomena. A relatively highly studied phenomenon is syntax, which is investigated both for earlier LSTM-based models (Linzen, Dupoux, and Goldberg 2016; Kuncoro et al. 2018) as well as for the more recent Transformer-based ones (Goldberg 2019; Hewitt and Manning 2019; Saphra and Lopez 2019; Jawahar, Sagot, and Seddah 2019; van Schijndel, Mueller, and Linzen 2019; Tenney et al. 2019). A recent work in this context is the **probe** proposed by Hewitt and Manning (2019), which enabled them to show that Transformer-based models encode human-like parse trees to a very good extent. In terms of semantics, fewer studies exist, including the probing study of Ettinger (2020) on semantic roles, and that of Tenney, Das, and Pavlick (2019), which also investigates entity types and relations. The closest analysis to ours is that of Peters et al. (2018b), which provides a deep analysis of contextualized word embeddings, both from the representation point of view and per architectural choices. In the same spirit, Conneau et al. (2018) proposed a number of linguistic probing tasks to analyze sentence embedding models. Perhaps more related to the topic of this article, Shwartz and Dagan (2019) showed how contextualized embeddings are able to capture non-literal usages of words in the context of lexical composition. For a complete overview of existing probe and analysis methods, the survey of Belinkov and Glass (2019) provides a synthesis of analysis studies on neural network methods. The more recent survey of Rogers, Kovaleva, and Rumshisky (2020) is a similar synthesis but targeted at BERT and its derivatives.

Despite all this analytical work, the investigation of neural language models from the perspective of ambiguity (and, in particular, lexical ambiguity) has been surprisingly neglected. In the following we discuss studies that aimed at shedding some light on this important linguistic phenomenon.

2.2 Lexical Ambiguity and Language Models

Given its importance, lexical ambiguity has for long been an area of investigation in vector space model representations (Schütze 1993; Reisinger and Mooney 2010; Camacho-Collados and Pilehvar 2018). In a recent study on word embeddings, Yaghoobzadeh et al. (2019) showed that Word2vec (Mikolov et al. 2013a) can effectively capture different coarse-grained senses if they are all frequent enough and evenly distributed. In this work we try to extend this conclusion to a language model-based representation and to the more realistic scenario of disambiguating words in context, rather than probing them in isolation for if they capture specific senses (as was the case in that work).

Most of the works analyzing language models and lexical ambiguity have opted for lexical substitution as their experimental benchmark. Amrami and Goldberg (2018) showed that an LSTM language model can be effectively applied to the task of word sense induction. In particular, they analyzed how the predictions of an LSTM for a word in context provided a useful way to retrieve substitutes, proving that this information is indeed captured in the language model. From a more analytical point of view, Aina, Gulordava, and Boleda (2019) proposed a probe task based on lexical substitution to understand the internal representations of an LSTM language model for predicting

words in context. Similarly, Soler et al. (2019) provided an analysis of LSTM-based contextualized embeddings in distinguishing between usages of words in context. As for Transformer-based models, Zhou et al. (2019) proposed a model based on BERT to achieve state-of-the-art results in lexical substitution, showing that BERT is particularly suited to find senses of a word in context. While lexical substitution has been shown to be an interesting proxy for WSD, we provide a direct and in-depth analysis of the explicit capabilities of recent language models in encoding lexical ambiguity, both quantitatively and qualitatively.

Another related work to ours is the analysis of Reif et al. (2019) on quantifying the geometry of BERT. The authors observed that, generally, when contextualized BERT embeddings for ambiguous words are visualized, clear clusters for different senses are identifiable. They also devised an experiment to highlight a potential failure with BERT (or presumably other attention-based models): It does not necessarily respect semantic boundaries when attending to neighboring tokens. In our qualitative analysis in Section 6.4 we further explore this. Additionally, Reif et al. (2019) present evidence supporting the specialization of representations from intermediate layers of BERT for sense representation, which we further confirm with layer-wise WSD evaluation in Section 3.4.5. Despite these interesting observations, their paper mostly focuses on the syntactic properties of BERT, similarly to most other studies in the domain (see Section 2.1).

Finally, a few works have attempted to induce semantic priors coming from knowledge resources like WordNet to improve the generalization of pretrained language models like BERT (Levine et al. 2020; Peters et al. 2019). Other works have investigated BERT's emergent semantic space using clustering analyses (Yenicelik, Schmidt, and Kilcher 2020; Chronis and Erk 2020), seeking to characterize how distinct sense-specific representations occupy this space.

Our work differs in that we are trying to understand to what extent pretrained language models already encode this semantic knowledge and, in particular, what are their implicit practical disambiguation capabilities.

2.3 Evaluation Benchmarks

The most common evaluation benchmarks for WSD are based on fine-grained resources, with WordNet (Fellbaum 1998) being the de facto sense inventory. For example, the unified all-words WSD benchmark of Raganato, Camacho-Collados, and Navigli (2017) is composed of five data sets from Senseval/SemEval tasks: Senseval-2 (Edmonds and Cotton 2001, SE02), Senseval-3 (Mihalcea, Chklovski, and Kilgarriff 2004, SE03), SemEval-2007 (Agirre, Màrquez, and Wicentowski 2007, SE07), SemEval-2013 (Navigli, Jurgens, and Vannella 2013, SE13), and SemEval-2015 (Moro and Navigli 2015, SE15). Vial, Lecouteux, and Schwab (2018) extended this framework with other manually and automatically constructed data sets.[2] All these data sets are WordNet-specific and mostly use SemCor (Miller et al. 1993) as their training set. Despite being the largest WordNet-based sense-annotated data set, SemCor does not cover many senses occurring in the test sets, besides providing a limited number of examples per sense. Although scarcity in the training data is certainly a realistic setting, in this article we are interested in analyzing the limits of language models with and without training data, also for senses not included in WordNet, and run a qualitative analysis.

2 Pasini and Camacho-Collados (2020) provide an overview of existing sense-annotated corpora for WordNet and other resources.

To this end, in addition to running an evaluation in standard benchmarks, for this article we constructed a coarse-grained WSD data set, called CoarseWSD-20. CoarseWSD-20 includes a selection of 20 ambiguous words of different nature (see Section 4 for more details on CoarseWSD-20) where we run a qualitative analysis on various aspects of sense-specific information encoded in language models. Perhaps the closest data sets to CoarseWSD-20 are those of Lexical Sample WSD (Edmonds and Cotton 2001; Mihalcea, Chklovski, and Kilgarriff 2004; Pradhan et al. 2007). These data sets usually target dozens of ambiguous words and list specific examples for their different senses. However, these examples are usually fine-grained, limited in number,[3] and are limited to concepts (i.e., no entities such as *Java* are included). The CoarseWSD-20 data set is similar in spirit, but has larger training sets extracted from Wikipedia. Constructing the data set based on the sense inventory of Wikipedia brings the additional advantage of having both entities and concepts as targets, and a direct mapping to Wikipedia pages, which is the most common resource for entity linking (Ling, Singh, and Weld 2015; Usbeck et al. 2015), along with similar inter-connected resources such as DBpedia.

Another related data set to CoarseWSD-20 is WIKI-PSE (Yaghoobzadeh et al. 2019). Similarly to ours, WIKI-PSE is constructed based on Wikipedia, but with a different purpose. WIKI-PSE clusters all Wikipedia concepts and entities into eight general "semantic classes." This is an extreme coarsening of the sense inventory that may not fully reflect the variety of human-interpretable senses that a word has. Instead, for CoarseWSD-20, sense coarsening is performed at the word level, which preserves sense-specific information. For example, the word *bank* in WIKI-PSE is mainly identified as a *location* only, conflating the financial institution and river meanings of the word, whereas CoarseWSD-20 distinguishes between the two senses of *bank*. Moreover, our data set is additionally post-processed in a semi-automatic manner (an automatic preprocessing, followed by a manual check for problematic cases), which helps remove errors from the Wikipedia dump.

3. Word Sense Disambiguation: An Overview

Our analysis is focused on the task of word sense disambiguation. WSD is a core module of human cognition and a long-standing task in NLP. Formally, given a word in context, the task of WSD consists of selecting the intended meaning (sense) from a predefined set of senses for that word defined by a sense inventory (Navigli 2009). For example consider the word *star* in the following context:

- Sirius is the brightest *star* in Earth's night.

The task of a WSD system is to identify that the usage of *star* in this context refers to its astronomical meaning (as opposed to celebrity or star shape, among others). The context could be a document, a sentence, or any other information-carrying piece of text that can provide a hint on the intended semantic usage,[4] probably as small as a word, for example, "*dwarf* star."[5]

3 For instance, the data set of Pradhan et al. (2007), which is the most recent and the largest among the three mentioned lexical sample data sets, provides an average of 320/50 training/test instances for each of the 35 nouns in the data set. In contrast, CoarseWSD-20 includes considerably larger data sets for all words (1,160 and 510 sentences on average for each word in the training and test sets, respectively).

4 For this analysis we focus on sentence-level WSD, because it is the most standard practice in the literature.

5 A *dwarf star* is a relatively small star with low luminosity, such as the Sun.

WSD is described as an AI-hard[6] problem (Mallery 1988). In a comprehensive survey of WSD, Navigli (2009) discusses some of the reasons behind its difficulty, including heavy reliance on knowledge, difficulty in distinguishing fine-grained sense distinctions, and lack of application to real-world tasks. On WordNet-style sense inventories, the human-level performance (which is usually quoted as glass ceiling) is estimated to be 80% in the fine-grained setting (Gale, Church, and Yarowsky 1992a) and 90% for the coarse-grained one (Palmer, Dang, and Fellbaum 2007). This performance gap can be mainly attributed to the fine-grained semantic distinctions in WordNet that are sometimes even difficult for humans to distinguish. For instance, the noun *star* has 8 senses in WordNet 3.1, two of which refer to the astronomical sense (celestial body) with the minor semantic difference of if the star is visible from Earth at night. In fact, it is argued that sense distinctions in WordNet are too fine-grained for many NLP applications (Hovy, Navigli, and Ponzetto 2013). CoarseWSD-20 addresses this issue by devising sense distinctions that are easily interpretable by humans, essentially pushing the human performance on the task.

Similarly to many other tasks in NLP, WSD has gone under significant change after the introduction of Transformer-based language models, which are now dominating most WSD benchmarks. In the following we first present a background on existing sense inventories, with a focus on WordNet (Section 3.1), and then describe the state of the art in both the conventional paradigm (Section 3.2) and the more recent paradigm based on (Transformer-based) language models (Section 3.3). We then carry out a quantitative evaluation of some of the most prominent WSD approaches in each paradigm in various disambiguation scenarios, including fine- and coarse-grained settings (Section 3.4). This quantitative analysis is followed by an analysis of layer-wise representations (Section 3.4.5) and performance per word categories (parts of speech, Section 3.4.6).

3.1 Sense Inventories

Given that WSD is usually tied with sense inventories, we briefly describe existing sense inventories that are also used in our experiments. The main sense inventory for WSD research in English is the Princeton WordNet (Fellbaum 1998). The basic constituents of this expert-made lexical resource are **synsets**, which are sets of synonymous words that represent unique concepts. A word can belong to multiple synsets denoting to its different meanings. Version 3.0 of the resource, which is used in our experiments, covers 147,306 words and 117,659 synsets.[7] WordNet is also available for languages other than English through the Open Multilingual WordNet project (Bond and Foster 2013) and related efforts.

Other common-sense inventories are Wikipedia and BabelNet. The former is generally used for Entity Linking or **Wikification** (Mihalcea and Csomai 2007), in which the Wikipedia pages are considered as concept or entities to be linked in context. On the other hand, BabelNet (Navigli and Ponzetto 2012) is a merger of WordNet, Wikipedia, and several other lexical resources, such as Wiktionary and OmegaWiki. One of the key

6 By analogy to NP-completeness, the most difficult problems are referred to as AI-complete, implying that solving them is equivalent to solving the central artificial intelligence problem.

7 There are several other variants of WordNet available, either the newer v3.1, which is slightly different from the former version, or other non-Princeton versions that improve coverage, such as WordNet 2020 (McCrae et al. 2020) or CROWN (Jurgens and Pilehvar 2015). We opted for v3.0 given that it is the widely used inventory according to which most existing benchmarks are annotated.

features of this resource is its multilinguality, highlighted by the 500 languages covered in its most recent release (version 5.0).

3.2 WSD Paradigms

WSD approaches are traditionally categorized as **knowledge-based** and **supervised**. The latter makes use of sense-annotated data for its training whereas the former exploits sense inventories, such as WordNet, for the encoded knowledge, such as sense glosses (Lesk 1986; Banerjee and Pedersen 2003; Basile, Caputo, and Semeraro 2014), semantic relations (Agirre, de Lacalle, and Soroa 2014; Moro, Raganato, and Navigli 2014), or sense distributions (Chaplot and Salakhutdinov 2018). Supervised WSD has been shown to clearly outperform the knowledge-based counterparts, even before the introduction of pretrained language models (Raganato, Camacho-Collados, and Navigli 2017). Large pretrained language models have further provided improvements, with BERT-based models currently approaching human-level performance (Loureiro and Jorge 2019a; Vial, Lecouteux, and Schwab 2019; Huang et al. 2019; Bevilacqua and Navigli 2020; Blevins and Zettlemoyer 2020). A third category of WSD techniques, called **hybrid**, has recently attracted more attention. In this approach, the model benefits from both sense-annotated instances and knowledge encoded in sense inventories.[8] Most of the recent state-of-the-art approaches can be put in this category.

3.3 Language Models for WSD

In the context of Machine Translation (MT), a language model is a statistical model that estimates the probability of a sequence of words in a given language. Recently, the scope of LMs has gone far beyond MT and generation tasks. This is partly due to the introduction of Transformers (Vaswani et al. 2017), attention-based neural architectures that have proven immense potential in capturing complex and nuanced linguistic knowledge. In fact, despite their recency, Transformer-based LMs dominate most language understanding benchmarks, such as GLUE (Wang et al. 2018) and SuperGLUE (Wang et al. 2019).

There are currently two popular varieties of Transformer-based Language Models (LMs), differentiated most significantly by their choice of language modeling objective. There are causal (or left-to-right) models, epitomized by GPT-3 (Brown et al. 2020), where the objective is to predict the next word, given the past sequence of words. Alternatively, there are masked models, where the objective is to predict a masked (i.e., hidden) word given its surrounding words, traditionally known as the Cloze task (Taylor 1953), of which the most prominent example is BERT. Benchmark results reported in Devlin et al. (2019) and Brown et al. (2020) show that masked LMs are preferred for semantic tasks, whereas causal LMs are more suitable for language generation tasks. As a potential explanation for the success of BERT-based models, Voita, Sennrich, and Titov (2019) present empirical evidence suggesting that the masked LM objective induces models to produce more generalized representations in intermediate layers.

In our experiments, we opted for the BERT (Devlin et al. 2019) and ALBERT (Lan et al. 2020) models given their prominence and popularity. Nonetheless, our empirical

8 Note that knowledge-based WSD systems might benefit from sense frequency information obtained from sense-annotated data, such as SemCor. Given that such models do not incorporate sense-annotated instances, we do not categorize them as hybrid.

analysis could be applied to other pretrained language models as well (e.g., Liu et al. 2019b; Raffel et al. 2020). Our experiments focus on two dominant WSD approaches based on language models: (1) Nearest Neighbors classifiers based on features extracted from the model (Section 3.3.1), and (2) fine-tuning of the model for WSD classification (Section 3.3.2). In the following we describe the two strategies.

3.3.1 Feature Extraction. Neural LMs have been utilized for WSD, even before the introduction of Transformers, when LSTMs were the first choice for encoding sequences (Melamud, Goldberger, and Dagan 2016; Yuan et al. 2016; Peters et al. 2018a). In this context, LMs were often used to encode the context of a target word, or in other words, generate a contextual embedding for that word. Allowing for various sense-inducing contexts to produce different word representations, these contextual embeddings proved more suitable for lexical ambiguity than conventional word embeddings (e.g., Word2vec).

Consequently, Melamud, Goldberger, and Dagan (2016), Yuan et al. (2016), and Peters et al. (2018a) independently demonstrated that, given sense-annotated corpora (e.g., SemCor), it is possible to compute an embedding for a specific word sense as the average of its contextual embeddings. Sense embeddings computed in this manner serve as the basis for a series of WSD systems. The underlying approach is straightforward: Match the contextual embedding of the word to be disambiguated against its corresponding pre-computed sense embeddings. The matching is usually done using a simple k Nearest Neighbors (NN) (often with $k = 1$) classifier; hence, we refer to this feature extraction approach as **1NN** in our experiments. A simple 1NN approach based on LSTM contextual embeddings proved effective enough to rival the performance of other systems using task-specific training, such as Raganato, Delli Bovi, and Navigli (2017), despite using no WSD specific modeling objectives. Loureiro and Jorge (2019a, LMMS) and Wiedemann et al. (2019) independently showed that the same approach using contextual embeddings from BERT could in fact surpass the performance of those task-specific alternatives. Loureiro and Jorge (2019a) also explored a propagation method using WordNet to produce sense embeddings for senses not present in training data (LMMS_{1024}) and a variant that introduced information from glosses into the same embedding space (LMMS_{2048}). Similar methods have been also introduced for larger lexical resources such as BabelNet, with similar conclusions (Scarlini, Pasini, and Navigli 2020a, SensEmBERT).

There are other methods based on feature extraction that do not use 1NN for making predictions. Vial, Lecouteux, and Schwab (2019, Sense Compression) used contextual embeddings from BERT as input for additional Transformer encoder layers with a softmax classifier on top. Blevins and Zettlemoyer (2020) also experimented with a baseline using the final states of a BERT model with a linear classifier on top. Finally, the solution by Bevilacqua and Navigli (2020) relied on an ensemble of sense embeddings from LMMS and SensEmBERT, along with additional resources, to train a high performance WSD classifier.

3.3.2 Fine-Tuning. Another common approach to benefiting from contextualized language models in downstream tasks is fine-tuning. For each target task, it is possible to simply plug in the task-specific inputs and outputs into pretrained models, such as BERT, and fine-tune all or part of the parameters end-to-end. This procedure adjusts the model's parameters according to the objectives of the target task, for example, the classification task in WSD. One of the main drawbacks of this type of supervised model is their need for building a model for each word, which is unrealistic in practice for

Computational Linguistics Volume 47, Number 2

all-words WSD. However, there are several successful WSD approaches in this category that overcome this limitation in different ways. GlossBERT (Huang et al. 2019) uses sense definitions to fine-tune the language model for the disambiguation task, similarly to a text classification task. KnowBERT (Peters et al. 2019) fine-tunes BERT for entity linking exploiting knowledge bases (WordNet and Wikipedia) as well as sense definitions. BEM (Blevins and Zettlemoyer 2020) proposes a bi-encoder method that learns to represent sense embeddings leveraging sense definitions while performing the optimization jointly with the underlying BERT model.

3.4 Evaluation in Standard Benchmarks

In our first experiment, we perform a quantitative evaluation on the unified WSD evaluation framework (Section 3.4.3), which verifies the extent to which a model can distinguish between different senses of a word as defined by WordNet's inventory.

3.4.1 BERT Models. For this task we use a NN strategy (1NN henceforth) that has been shown to be effective with pretrained language models, both for LSTMs and more recently for BERT (see Section 3.3.1). In particular, we used the cased base and large variants of BERT released by Devlin et al. (2019), as well as the xxlarge (v2) variant of ALBERT (Lan et al. 2020), via the Transformers framework (v2.5.1) (Wolf et al. 2020). Following LMMS, we also average sub-word embeddings and represent contextual embeddings as the sum of the corresponding representations from the final four layers. However, here we do not apply the LMMS propagation method aimed at fully representing the sense inventory, resorting to the conventional MFS fallback for lemmas unseen during training.

3.4.2 Comparison Systems. In addition to BERT and ALBERT, we include results for 1NN systems that exploit precomputed sense embeddings, namely, Context2vec (Melamud, Goldberger, and Dagan 2016) and ELMo (Peters et al. 2018a). Moreover, we include results for hybrid systems, namely, supervised models that also make use of additional knowledge sources (cf. Section 3.2), particularly semantic relations and textual definitions in WordNet. Besides the models already discussed in Sections 3.3.1 and 3.3.2, we also report results from additional hybrid models. Raganato, Delli Bovi, and Navigli (2017, Seq2Seq) trained a neural BiLSTM sequence model with losses specific not only to specific senses from SemCor but also part-of-speech tags and WordNet supersenses. EWISE (Kumar et al. 2019), which inspired EWISER (Bevilacqua and Navigli 2020), also uses a BiLSTM to learn contextual representations that can be matched against sense embeddings learned from both sense definitions and semantic relations.

For completeness we also add some of the best linear supervised baselines, namely, IMS (Zhong and Ng 2010) and IMS with embeddings (Zhong and Ng 2010; Iacobacci, Pilehvar, and Navigli 2016, IMS+emb), which are Support Vector Machine (SVM) classifiers based on several manually curated features. Finally, we report results for knowledge-based systems (KB) that mainly rely on WordNet: Lesk_{ext}+emb (Basile, Caputo, and Semeraro 2014), Babelfy (Moro, Raganato, and Navigli 2014), UKB (Agirre, López de Lacalle, and Soroa 2018), and TM (Chaplot and Salakhutdinov 2018). More recently, SyntagRank (Scozzafava et al. 2020) showed best KB results by combining WordNet with the SyntagNet (Maru et al. 2019) database of syntagmatic relations. However, as discussed in Section 3.2, we categorize these as knowledge-based because they do not directly incorporate sense-annotated instances as their source of knowledge.

Table 1
F-Measure performance on the unified WSD evaluation framework (Raganato, Camacho-Collados, and Navigli 2017) for three classes of WSD models (i.e., knowledge-based [KB], supervised, and hybrid), and for two sense specification settings (i.e., fine-grained [FN] and coarse-grained [CS]. Results marked with * make use of SE07/SE15 as development set. Systems marked with † rely on external resources other than WordNet. The results from complete rows were computed by ourselves given the system outputs, while those from incomplete rows were taken from the original papers.

	Type	System	SE2		SE3		SE07		SE13		SE15		ALL	
			FN	CS	FN	CS	FN	CS	FN	CS	FN	CS	FN	CS
KB		$Lesk_{ext}$+emb	63.0	74.9	63.7	75.5	56.7	71.6	66.2	77.4	64.6	73.9	63.7	75.3
		Babelfy†	67.0	78.4	63.5	77.5	51.6	68.8	66.4	77.0	70.3	79.1	65.5	77.3
		TM	69.0	—	66.9	—	55.6	—	65.3	—	69.6	—	66.9	—
		UKB	68.8	81.2	66.1	78.1	53.0	70.8	68.8	79.1	70.3	77.4	67.3	78.7
		SyntagRank	**71.6**	—	**72.0**	—	**59.3**	—	**72.2**	—	**75.8**	—	**71.7**	—
Supervised	**SVM**	IMS	70.9	81.5	69.3	80.8	61.3	74.3	65.3	77.4	69.5	75.7	68.4	79.1
		IMS+emb	72.2	82.8	70.4	81.5	62.6	75.8	65.9	76.9	71.5	76.7	69.6	79.8
	1NN	Context2vec	71.8	82.6	69.1	80.5	61.3	74.5	65.6	78.0	71.9	76.6	69.0	79.7
		ELMo	71.6	82.8	69.6	80.9	62.2	74.7	66.2	77.7	71.3	77.0	69.0	79.6
		BERT-Base	75.5	84.9	71.5	81.4	65.1	78.9	69.8	82.1	73.4	78.1	72.2	82.0
		BERT-Large	76.3	84.8	**73.2**	**82.9**	66.2	80.0	71.7	83.1	74.1	**79.1**	73.5	82.8
		ALBERT-XXL	**76.6**	**85.6**	73.1	82.6	**67.3**	**80.1**	**71.8**	**83.5**	**74.3**	78.3	**73.7**	**83.0**
Hybrid		Seq2Seq Att+Lex+PoS	70.1	—	68.5	—	63.1*	—	66.5	—	69.2	—	68.6*	—
		Sense Compr. Ens.	79.7	—	77.8	—	73.4	—	78.7	—	82.6	—	**79.0**	—
		LMMS 1024	75.4	—	74.0	—	66.4	—	72.7	—	75.3	—	73.8	—
		LMMS 2048	76.3	84.5	75.6	85.1	68.1	81.3	75.1	86.4	77.0	80.8	75.4	84.4
		EWISE	73.8	—	71.1	—	67.3*	—	69.4	—	74.5	—	71.8*	—
		KnowBert† WN+WK	76.4	85.6	76.0	85.1	71.4	82.6	73.1	83.8	75.4	80.2	75.1	84.1
		GlossBERT	77.7	—	75.2	—	72.5*	—	76.1	—	80.4	—	77.0*	—
		BEM	79.4	—	77.4	—	74.5*	—	79.7	—	**81.7**	—	79.0*	—
		EWISER†	**80.8**	—	**79.0**	—	**75.2**	—	**80.7**	—	81.8*	—	80.1*	—
	—	*MFS Baseline*	65.6	77.4	66.0	77.8	54.5	70.6	63.8	74.8	67.1	75.3	64.8	76.2

3.4.3 Data Sets: Unified WSD Benchmark. Introduced by Raganato, Camacho-Collados, and Navigli (2017) as an attempt to construct a standard evaluation framework for WSD, the unified benchmark comprises five data sets from Senseval/SemEval workshops (see Section 2.3).[9] The framework provides 7,253 test instances for 4,363 sense types. In total, around 3,663 word types are covered with an average polysemy of 6.2 and across four parts of speech: nouns, verbs, adjectives, and adverbs.

Note that the data sets are originally designed for the fine-grained WSD setting. Nonetheless, in addition to the fine-grained setting, we provide results on the coarse-grained versions of the same test sets. To this end, we merged those senses that belonged to the same domain according to CSI (Coarse Sense Inventory) domain labels from Lacerra et al. (2020).[10] With this coarsening, we can provide more meaningful comparisons and draw interpretable conclusions. Finally, we followed the standard procedure and trained all models on SemCor (Miller et al. 1993).

3.4.4 Results. Table 1 shows the results[11] of all comparison systems on the unified WSD framework, both for fine-grained (FN) and coarse-grained (CS) versions. The $LMMS_{2048}$

9 Data set downloaded from `http://lcl.uniroma1.it/wsdeval/`.
10 CSI domains downloaded from `http://lcl.uniroma1.it/csi`.
11 SensEmBERT not included because it is only applicable to the noun portions of these test sets.

Computational Linguistics Volume 47, Number 2

hybrid model, which is based on the 1NN BERT classifier, is the best-performer based solely on feature extraction. The latest fine-tuning hybrid solutions, particularly BEM and EWISER, show overall best performance, making the case for leveraging glosses and semantic relations to optimize pretrained weights for the WSD task. Generally, all BERT-based models achieve fine-grained results that are in the same ballpark as human average inter-annotator agreements for fine-grained WSD, which ranges from 64% and 80% in the three earlier data sets of this benchmark (Navigli 2009). In the more interpretable coarse-grained setting, LMMS achieves a score of 84.4%, similar to the other BERT-based models, which surpass 80%. The remaining supervised models perform roughly equal, marginally below 80% and clearly underperformed by BERT-based models.

3.4.5 Layer Performance. Current BERT-based 1NN WSD methods (see Section 3.3.1), such as LMMS and SensEmBERT, apply a **pooling** procedure to combine representations extracted from various layers of the model. The convention is to sum the embeddings from the last four layers, following the Named Entity Recognition experiments reported by Devlin et al. (2019). It is generally understood that lower layers are closer to their static representations (i.e., initialization) and, conversely, upper layers better match the modeling objectives (Tenney, Das, and Pavlick 2019). Still, Reif et al. (2019) have shown that this relation is not monotonic when it comes to sense representations from BERT. Additional probing studies have also pointed to irregular progression of context-specificity and token identity across the layers (Ethayarajh 2019; Voita, Sennrich, and Titov 2019), two important pre-requisites for sense representation.

Given our focus on measuring BERT's adeptness for WSD, and the known variability in layer specialization, we performed an analysis to reveal which layers produce representations that are most effective for WSD. This analysis involved obtaining sense representations learned from SemCor for each layer individually using the process described in Section 3.3.1.

Figure 1 shows the performance of each layer using a restricted version of the MASC corpus (Ide et al. 2008) as a validation set where only annotations for senses that occur in SemCor are considered. Any sentence that contained annotations for senses not occurring in SemCor was removed, restricting this validation set to 14,645 annotations out of 113,518. We restrict the MASC corpus so that our analysis is not affected by strategies for inferring senses (e.g., Network Propagation) or fallbacks (e.g., Most Frequent Sense). This restricted version of MASC is based on the release introduced in Vial, Lecouteux, and Schwab (2018), which mapped annotations to Princeton WordNet (3.0).

Similarly to Reif et al. (2019), we find that lower layers are not as effective for disambiguation as upper layers. However, our experiment specifically targets WSD and its results suggest a different distribution of the best performing layers than those reported by Reif et al. (2019). Nevertheless, this analysis shows that the current convention of using the sum of the last four layers for sense representations is sensible, even if not optimal.

Several model probing works have revealed that the scalar mixing method introduced by Peters et al. (2018a) allows for combining information from all layers with improved performance on lexico-semantic tasks (Liu et al. 2019a; Tenney et al. 2019; de Vries, van Cranenburgh, and Nissim 2020). However, scalar mixing essentially involves training a learned probe, which can limit attempts at analyzing the inherent semantic space represented by NLMs (Mickus et al. 2020).

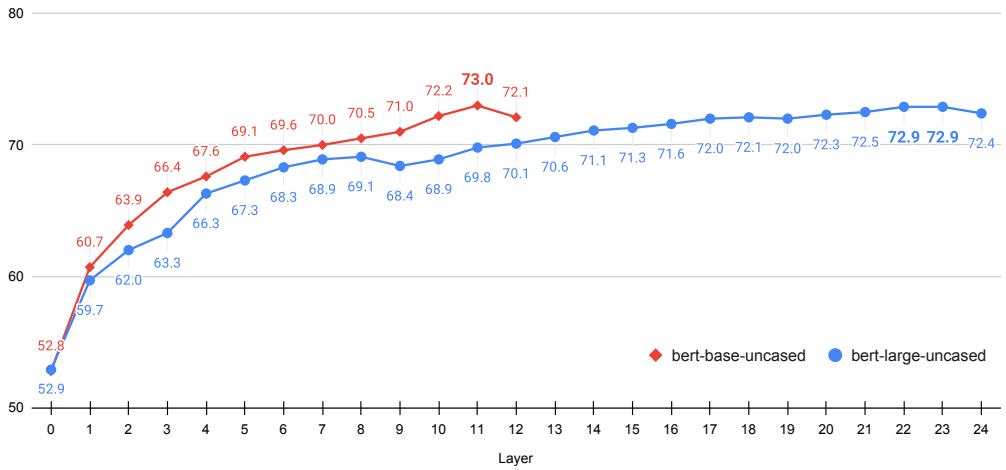


Figure 1
F-measure performance on a restricted version of the MASC corpus (Ide et al. 2008) for representations derived from individual layers of the two BERT models used in our experiments.

Table 2
F-Measure performance in the concatenation of all data sets of the unified WSD evaluation framework (Raganato, Camacho-Collados, and Navigli 2017), split by part of speech. As in Table 1 systems marked with † make use of external resources other than WordNet.

	Type	System	Nouns		Verbs		Adjectives		Adverbs	
			FN	CS	FN	CS	FN	CS	FN	CS
KB		UKB	71.2	80.5	50.7	69.2	75.0	82.7	77.7	91.3
		Lesk$_{ext}$+emb	69.8	79.0	51.2	69.2	51.7	62.4	80.6	92.8
		Babelfy†	68.6	78.9	49.9	67.6	73.2	82.1	79.8	91.6
Supervised	**1NN**	Context2vec	71.0	80.5	57.6	72.9	75.2	83.1	82.7	92.5
		ELMo	70.9	80.0	57.3	73.5	77.4	85.4	82.4	92.8
		BERT-Base	74.0	83.0	61.7	75.3	77.7	84.9	**85.8**	93.9
		BERT-Large	75.1	83.7	63.2	76.6	79.5	85.4	85.3	**94.2**
	SVM	IMS	70.4	79.4	56.1	72.5	75.6	84.1	82.9	93.1
		IMS+emb	71.9	80.5	56.9	73.1	75.9	83.8	84.7	93.4
Hybrid		LMMS$_{2048}$	**78.0**	**86.2**	64.0	76.5	**80.7**	**86.7**	83.5	92.8
		KnowBert† WN+WK	77.0	85.0	**66.4**	**78.8**	78.3	86.1	84.7	93.9
	–	*MFS Baseline*	67.6	77.0	49.6	67.2	73.1	82.0	80.5	92.9

3.4.6 Analysis by part of speech. Table 2 shows the results of BERT and the comparison systems by part of speech.[12] The results clearly show that verbs are substantially more difficult to model, which corroborates the findings of Raganato, Camacho-Collados, and Navigli (2017), while adverbs are the least problematic in terms of disambiguation. For example, in the fine-grained setting, BERT-Large achieves an overall F1 of 75.1% on nouns vs. 63.2% on verbs (85.3% on adverbs). The same trend is observed for other

12 For this table we only included systems for which we received access to their system outputs.

models, including hybrid ones. This may also be related to the electrophysiological evidence suggesting that humans process nouns and verbs differently (Federmeier et al. 2000). Another more concrete reason for this gap is due to the fine granularity of verb senses in WordNet. For instance, the verb *run* has 41 sense entries in WordNet, twelve of which denote some kind of motion.

The coarsening of sense inventory does help in bridging this gap, with the best models performing in the 75% ballpark. Nonetheless, the lower performance is again found in verb instances, with noun, adjective, and adverb performance being above 80% on the BERT-based models (above 90% in the case of adverbs). One problem with the existing coarsening methods is that they usually exploit domain-level information, whereas in some cases verbs do not belong to clear domains. For our example verb *run*, some of the twelve senses denoting motion are clustered into different domains, which eases the task for automatic models due to having fewer number of classes. However, one could argue that this clustering is artificial as all senses of the verb belong to the same domain.

Indeed, while the sense clustering provided by CSI (Lacerra et al. 2020) covers all PoS categories, it extends BabelDomains (Camacho-Collados and Navigli 2017), a domain clustering resource that covers mainly nouns. Although out of scope for this article, in the future it would be interesting to investigate verb-specific clustering methods (e.g., Peterson and Palmer 2018).

In the remainder of this article we focus on noun ambiguity, and check the extent to which language models can solve coarse-grained WSD in ideal settings. In Section 7, we extend the discussion about sense granularity in WSD.

4. CoarseWSD-20 Data Set

Standard WSD benchmarks mostly rely on WordNet. This makes the evaluations carried out on these data sets and the conclusions drawn from them specific to this resource only. Moreover, sense distinctions in WordNet are generally known to be too fine-grained (see more details about the fine granularity of WordNet in the discussion of Section 7) and annotations are scarce given the knowledge-acquisition bottleneck (Gale, Church, and Yarowsky 1992a; Pasini 2020). This prevents us from testing the limits of language models in WSD, which is one of the main motivations of this article.

To this end, we devise a new data set, CoarseWSD-20 henceforth, in an attempt to solve the aforementioned limitations. CoarseWSD-20 aims to provide a benchmark for the qualitative analysis of certain types of easily interpretable sense distinctions. Our data set also serves as a tool for testing the limits of WSD models in ideal training scenarios (i.e., with plenty of training data available per word).

In the following we describe the procedure we followed to construct CoarseWSD-20 (Section 4.1). Then, we present an estimation of the human performance (Section 4.2) and outline some relevant statistics (Section 4.3). Finally, we discuss the out-of-domain test set we built as a benchmark for experiments in Section 5.3.

4.1 Data Set Construction

CoarseWSD-20 targets noun ambiguity[13] for which, thanks to Wikipedia, data is more easily available. The data set focuses on the coarse-grained disambiguation setting,

13 There are arguably more types of ambiguity, including word categories (e.g., *play* as a noun or as a verb). Nevertheless, this type of ambiguity can be solved to a good extent by using state-of-the-art PoS taggers, which are able to achieve performances above 97% for English in general settings (Akbik, Blythe, and Vollgraf 2018).

which is more interpretable by humans (Lacerra et al. 2020). To this end, 20 words[14] and their corresponding senses were selected by a group of two expert computational linguists in order to provide a diverse data set. Wikipedia[15] was used as reference inventory and corpus. In this case, each Wikipedia page corresponds to an unambiguous sense. Sentences where a given Wikipedia page is referred to via a hyperlink are considered to be its corresponding sense-annotated sentences. The process to select 20 ambiguous words and their corresponding sense-annotated sentences was as follows:

1. A larger set of a few hundred ambiguous words that had a minimum of 30 occurrences[16] (i.e., sentences where one of their senses is referred to via a hyperlink) was selected.
2. Two experts selected 20 words based on a variety of criteria: type of ambiguity (e.g., spanning across domains or not), polysemy, overall frequency, distribution of instances across senses of the word, and interpretability. This process was performed semi-automatically, as initially the experts filtered words and senses manually providing a reduced set of words and associated senses. The main goal of this filtering was to discard those senses that were not easily interpretable or distinguishable by humans.

Once these 20 words were selected, we tokenized and lowercased the English Wikipedia and extracted all sentences that contained them and their selected senses as hyperlinks. All sentences were then semi-automatically verified so as to remove duplicate and noisy sentences. Finally, for each word we created a single data set based on a standard 60/40 train/test split.

4.2 Human Performance Estimation

As explained earlier this WSD data set was designed to be simple for humans to annotate. In other words, the senses considered for CoarseWSD-20 are easily interpretable. As a sanity check, we performed a disambiguation exercise with 1,000 instances randomly sampled from the test set (50 for each word). Four annotators[17] were asked to disambiguate a given target word in context using the CoarseWSD-20 sense inventory. Each annotator completed the task for five words. In the following section we provide details of the results of this annotation exercise, as well as general statistics of CoarseWSD-20.

4.3 Statistics

Table 3 shows the list of words, their associated senses, and the frequency of each word sense in CoarseWSD-20, along with the ratio of the first sense with respect to

14 The main justification to select 20 words (and no more) was the extent of experiments and the computation required to run a deep qualitative analysis (see Section 5.1). A larger number of words would have prevented us from running the analyses at the depth we envisaged: 20 provided a good trade-off between having a heterogeneous set of words and a deep qualitative analysis.

15 We used the Wikipedia dump of May 2016.

16 This threshold was selected for the goal of testing the language models under close-to-ideal conditions. A real setting should also include senses with even lower frequency, the so-called *long tail* (Ilievski, Vossen, and Schlobach 2018; Blevins and Zettlemoyer 2020), which would clearly harm automatic models.

17 All annotators were fluent English speakers and understood the predefined senses for their assigned words.

Computational Linguistics Volume 47, Number 2

Table 3
Target words and their associated senses, represented by their Wikipedia page title, with their overall associated frequency in CoarseWSD-20 (train/test). *F2R* denotes the ratio of instances for first sense to the rest, while *Ent.* is the normalized entropy of sense distribution. Moreover, the *Human* performance is reported in terms of accuracy.

Word	F2R	Ent.	Hum	Senses	Frequency
apple	1.6	0.96	100	apple_inc	1,466/634
				apple	892/398
arm	2.8	0.83	100	arm_architecture	311/121
				arm	112/43
bank	23.1	0.28	98	bank	1,061/433
				bank_(geography)	46/22
bass	2.9	0.67	90	bass_guitar	2,356/1,005
				bass_(voice_type)	609/298
				double_bass	208/88
bow	1.0	0.87	98	bow_ship	266/117
				bow_and_arrow	185/72
				bow_(music)	72/26
chair	1.4	0.91	98	chairman	156/88
				chair	115/42
club	0.9	0.85	86	club	186/108
				nightclub	148/73
				club_(weapon)	54/21
crane	1.3	0.99	98	crane_(machine)	211/81
				crane_(bird)	161/76
deck	8.4	0.37	96	deck_(ship)	152/92
				deck_(building)	18/7
digit	2.2	0.74	100	numerical_digit	47/33
				digit_(anatomy)	21/9
hood	1.6	0.88	98	hood_(comics)	105/47
				hood_(vehicle)	42/13
				hood_(headgear)	24/22
java	1.4	0.96	100	java	2,641/1,180
				java_(progr._lang.)	1,863/749
mole	0.4	0.93	98	mole_(animal)	148/77
				mole_(espionage)	120/44
				mole_(unit)	108/42
				mole_sauce	53/23
				mole_(architecture)	51/20
pitcher	355.7	0.04	100	pitcher	6,403/2,806
				pitcher_(container)	18/13
pound	6.2	0.48	100	pound_mass	160/87
				pound_(currency)	26/10
seal	0.5	0.87	100	pinniped	305/131
				seal_(musician)	267/106
				seal_(emblem)	265/114
				seal_(mechanical)	38/12
spring	0.9	0.91	100	spring_(hidrology)	516/236
				spring_(season)	389/148
				spring_(device)	159/73
square	1.1	0.83	96	square	264/103
				square_(company)	167/62
				town_square	56/29
				square_number	21/13
trunk	1.3	0.85	100	trunk_(botany)	93/47
				trunk_(automobile)	36/16
				trunk_(anatomy)	35/14
yard	5.3	0.62	100	yard	121/61
				yard_(sailing)	23/11

Table 4
Statistics of the out of domain data set. The two rightmost columns show the number of instances for each of the seven words and their distribution across senses.

	Polysemy	Normalized entropy	No. of instances	Sense distribution
bank	2	0.87	48	34/14
chair	2	0.47	40	4/36
pitcher	2	0.52	17	15/2
pound	2	0.43	46	42/4
spring	3	0.63	31	3/24/4
square	3	0.49	26	22/2/2
club	2	0.39	13	12/1

the rest (F2E), normalized entropy[18] (Ent.), and an estimation of the human accuracy (see Section 4.2). The number of senses per word varies from 2 to 5 (11 words with two associated senses, 6 with three, 2 with four, and 1 with five) while the overall frequency ranges from 110 instances (68 for training) for *digit* to 9,240 (6,421 for training) for *pitcher*. As for the human performance, we can see how annotators did not have special difficulty in assigning the right sense for each word in context. Annotators achieve an accuracy of over 96% in all cases except for a couple of senses with slightly finer-grained distinctions such as *club* and *bass*.

Normalized entropy ranges from 0.04 to 0.99 (higher entropy shows more balanced sense distribution). While some words contain a roughly balanced distribution of senses (e.g., *crane* or *java*), other words' distribution are highly skewed (see normalized entropy values, e.g., for *pitcher* or *bank*).

Finally, in the Appendix we include more information for each of the senses available in CoarseWSD-20, including definitions and an example sentence from the data set.

4.4 Out of Domain Test Set

The CoarseWSD-20 data set was constructed exclusively based on Wikipedia. Therefore, the variety of language present in the data set might be limited. To verify the robustness of WSD models in a different setting, we constructed an out-of-domain test set from existing WordNet-based data sets.

To construct this test set, we leveraged BabelNet mappings from Wikipedia to WordNet (Navigli and Ponzetto 2012) to link the Wikipedia-based CoarseWSD-20 to WordNet senses. After a manual verification of all senses, we retrieved all sentences containing one of the target words in either SemCor (Miller et al. 1993) or any of the Senseval/SemEval evaluation data sets from Raganato, Camacho-Collados, and Navigli (2017). Finally, we only kept those target words for which all the associated senses were present in the WordNet-based sense annotated corpora and occurred at least 10 times. This resulted in a test set with seven target words (i.e., bank, chair, pitcher, pound, spring, square, and club). Table 4 shows the relevant statistics of this out-of-domain test set.

18 Computed as $\sum f_i \, log(f_i)$ normalized by $log(n)$ where n is the number of senses.

Computational Linguistics Volume 47, Number 2

5. Evaluation

In this section we report on our quantitative evaluation in the coarse-grained WSD setting on CoarseWSD-20. We describe the experimental setting in Section 5.1 and then present the main results on CoarseWSD-20 (Section 5.2) and the out-of-domain test set (Section 5.3).

5.1 Experimental Setting

CoarseWSD-20 consists of 20 separate sets, each containing sentences for different senses of the corresponding target word. Therefore, the evaluation can be framed as a standard classification task for each word.

Given the classification nature of the CoarseWSD-20 data sets, we can perform experiments with our 1NN BERT system and compare it with a standard fine-tuned BERT model (see Section 3.3 for more details on the LM-based WSD approaches). Note that fine-tuning for individual target words results in many models (one per word). Therefore, this setup would not be computationally feasible in a general WSD setting, as the number of models would approach the vocabulary size. However, in our experiments we are interested in verifying the limits of BERT, without any other confounds or model-specific restrictions.

To ensure that our conclusions are generalizable, we also report 1NN and fine-tuning results using ALBERT. In spite of substantial operational differences, BERT and ALBERT have the most similar training objectives and tokenization methods out of several other prominent Transformer-based models (Yang et al. 2019; Liu et al. 2019b), thus being the most directly comparable. Given the similar performance between BERT-Large and ALBERT-XXLarge on the main CoarseWSD-20 data set, we proceed with further experiments using only BERT.

We also include two FastText linear classifiers (Joulin et al. 2017) as baselines: FTX-B (base model without pretrained embeddings) and FTX-C (using pretrained embeddings from Common Crawl). We chose FastText as the baseline given its efficiency and competitive results for sentence classification.

Configuration. Our experiments with BERT and ALBERT used the Transformers framework (v2.5.1) developed by Wolf et al. (2020), and we used the uncased pretrained base and large models released by Devlin et al. (2019) for BERT, and the xxlarge (v2) models released by Lan et al. (2020) for ALBERT. We use the uncased variants of Transformers models to match the casing in CoarseWSD-20 (except for ALBERT, which is only available in cased variants). Following previous feature extraction works (including our experiment in Section 3.4.1), with CoarseWSD-20 we also average sub-word representations and use the sum of the last four layers when extracting contextual embeddings. For fine-tuning experiments, we used a concatenation of the average embedding of target word's sub-words with the embedding of the [CLS] token, and fed them to a classifier. We used the same default hyper-parameter configuration for all the experiments. Given the fluctuation of results with fine-tuning, all the experiments are based on the average of three independent runs. Our experiments with FastText used the official package[19] (v0.9.1), with FastText-Base corresponding to the default supervised classification pipeline using randomly-initialized vectors, and FastText-Crawl corresponding to the

19 `https://fasttext.cc/`.

same pipeline but starting with pretrained 300-dimensional vectors based on Common Crawl. Following Joulin et al. (2017), classification with FastText is performed using multinominal logistic regression and averaged sub-word representations.

Evaluation Measures. In a classification setting, the performance of a model is measured by various metrics, among which precision, recall, and F-score are the most popular. Let TP_i (true-positive) and FP_i (false-positive) be the number of instances correctly / incorrectly classified as class c_i, respectively. Also, let TN_i (true-negative) and FN_i (false-negative) be the number of instances correctly / incorrectly classified as class c_j for any $j \neq i$. Therefore, for class c_i, precision P_i and recall R_i are defined as follows:

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad (1) \qquad R_i = \frac{TP_i}{TP_i + FN_i} \quad (2)$$

In other words, precision is the fraction of relevant instances among the retrieved instances, and recall is the fraction of the total number of relevant instances that were actually retrieved. The F-score F_i for class c_i is then defined as the harmonic mean of its precision and recall values:

$$F_i = \frac{2}{P_i^{-1} + R_i^{-1}} = 2\frac{P_i.R_i}{P_i + R_i} \quad (3)$$

In order to have a single value to measure the overall performance of the model, we can take the weighted average of these computed values over all the classes, which is referred to as average micro, if the weights are set to be the number of instances for each class, and macro if the weights are set to be equal. For our experiments we mainly report *Macro-F1* and *Micro-F1*.

Number of Experiments. To provide an idea of the experiments run on (including the analysis in Section 6), in the following we detail the number of computations required. We evaluated six models, each of them trained and tested separately for each word (there are twenty of them). The same models are also trained with balanced data sets (Section 6.2.1). In total, 240 models trained and tested for the main results (excluding multiple runs). Then, the computationally more demanding models (BERT-Large) are also evaluated on the out-of-domain test set, and trained with different training data sizes (Section 6.2.2) and with fixed number of examples (Section 6.3). In the latter case, BERT-base and FastText models are also considered (sometimes with multiple runs). As a rough estimate, all the experiments took over 1,500 hours on a Tesla K80 GPU. These experiments do not include the experiments run in the standard benchmarks (Section 3.4) and all the extra analyses and prior experimental tests that did not make it into the article.

5.2 Results

Word-specific results for different configurations of BERT and ALBERT as well as the FastText baseline are shown in Table 5. In general, results are high for all Transformer-based models, over 90% in most cases. This reinforces the potential of language models for WSD, both in its light-weight 1NN and in the fine-tuning settings. Although BERT-Large slightly improves over BERT-Base, the performance of the former is very

Table 5
Micro-F1 (top) and macro-F1 (bottom) performance on the full CoarseWSD-20 data set for eight different models: FastText-Base (FTX-B) and -Crawl (FTX-C), 1NN and fine-tuned BERT-Base (BRT-B), -Large (BRT-L), and ALBERT-XXL (ALBRT). An estimation of the human performance (see Section 4.2 for more details) and the most frequent sense (MFS) baseline are also reported for each word. Rows in each table are sorted by the entropy of sense distribution (see Table 3), in descending order. Table cells are highlighted (from red to green) for better interpretability.

Word	Human	MFS	Static emb.		1NN			Fine-tune		
			FTX-B	FTX-C	BRT-B	BRT-L	ALBRT	BRT-B	BRT-L	ALBRT
Micro-F1 (Accuracy)										
crane	*98.0*	51.6	91.7	94.9	93.6	96.8	98.1	97.5	98.1	96.8
java	*100.0*	61.2	98.8	99.4	99.6	99.6	99.6	99.7	99.7	99.5
apple	*100*	61.4	96.5	98.4	99.0	99.2	99.4	99.6	99.6	99.3
mole	*98.0*	37.4	87.4	93.2	97.1	98.5	98.1	98.9	98.9	98.5
spring	*100*	51.6	91.9	94.5	97.4	97.8	99.3	98.0	98.3	98.2
chair	*98.0*	67.7	81.5	88.5	96.2	96.2	95.4	96.7	96.2	94.1
hood	*98.0*	57.3	80.5	89.0	98.8	100	98.8	98.0	99.6	98.8
seal	*100*	36.1	88.7	95.0	96.4	98.1	97.5	99.0	99.0	98.3
bow	*98.0*	54.4	89.8	95.8	96.3	95.3	96.7	97.5	98.5	97.7
club	*86.0*	53.5	79.2	80.7	81.2	85.1	82.7	85.2	84.7	84.3
trunk	*100*	61.0	84.4	90.9	96.1	98.7	98.7	97.8	98.3	99.1
square	*96.0*	49.8	87.0	90.3	95.2	96.1	94.2	95.8	95.7	96.5
arm	*100*	73.8	94.5	98.2	99.4	99.4	99.4	99.4	99.4	99.6
digit	*100*	78.6	92.9	100.0	100.0	100.0	100.0	99.2	100.0	100.0
bass	*90.0*	72.3	93.9	94.2	80.7	84.5	85.5	95.5	95.8	95.7
yard	*100*	84.7	86.1	94.4	76.4	88.9	93.1	98.6	99.5	99.5
pound	*100*	89.7	87.6	87.6	86.6	89.7	95.9	94.9	94.9	96.6
deck	*96.0*	92.9	91.9	93.9	89.9	91.9	94.9	96.6	95.3	97.0
bank	*98.0*	95.2	96.9	98.0	99.6	99.8	99.8	99.6	99.3	99.3
pitcher	*100*	99.5	99.6	99.7	99.9	99.9	100.0	100.0	100.0	99.8
AVG		66.5	90.0	93.8	94.0	95.8	96.4	97.4	97.5	97.4
Macro-F1										
crane	–	34.0	91.7	94.8	93.5	96.7	98.1	97.5	98.1	96.8
java	–	38.0	98.7	99.4	99.7	99.6	99.6	99.7	99.7	99.5
apple	–	38.1	96.2	98.1	99.0	99.1	99.3	99.6	99.6	99.3
mole	–	10.9	84.4	91.0	97.6	99.0	98.4	98.9	99.2	98.8
spring	–	22.7	91.1	94.9	97.4	97.8	99.2	97.8	98.1	98.2
chair	–	40.4	79.5	86.5	94.7	94.7	94.7	96.1	95.5	93.3
hood	–	24.3	70.5	83.2	98.5	100.0	98.5	97.8	99.6	98.3
seal	–	13.3	72.7	92.6	97.3	98.5	98.1	98.9	98.6	97.9
bow	–	23.5	83.3	93.7	97.0	95.7	97.3	97.5	98.6	96.8
club	–	23.2	73.2	80.5	84.6	88.7	87.1	84.3	84.1	84.0
trunk	–	25.3	76.0	85.9	97.9	99.3	99.3	97.6	98.0	99.0
square	–	16.6	67.7	76.3	92.5	94.7	89.7	92.2	91.4	93.5
arm	–	42.5	92.5	98.0	99.6	99.6	99.6	99.2	99.2	99.5
digit	–	44.0	83.3	100.0	100.0	100.0	100.0	98.8	100.0	100.0
bass	–	28.0	80.2	81.3	79.1	84.0	87.1	87.5	87.6	86.9
yard	–	45.9	54.5	81.8	86.1	93.4	95.9	97.2	99.1	99.1
pound	–	47.3	48.9	53.3	92.5	94.3	97.7	84.4	83.9	90.4
deck	–	48.2	56.1	57.1	88.0	95.7	84.1	83.4	78.0	85.2
bank	–	48.8	68.2	79.5	95.5	97.7	97.7	97.9	95.6	96.3
pitcher	–	49.9	61.5	69.2	99.9	100.0	100.0	97.3	97.3	89.2
AVG	–	33.2	76.5	84.9	94.5	96.4	96.1	95.2	95.1	95.1

similar to that of ALBERT-XXL across different configurations, despite having different architectures, number of parameters, and training objectives. Overall, performance variations in different models are similar to those for the human baseline. For instance, words such as *java* and *digit* seem easy for both humans and models to disambiguate, whereas words such as *bass* and *club* are challenging perhaps because of their more fine-grained distinctions.[20] As a perhaps surprising result, having more training instances does not necessarily lead to better performance, indicated by the very low Pearson correlation (0.2 or lower) of the number of training instances with results in all BERT configurations. Also, higher polysemy is not a strong indicator of lower performance (see Table 4.3 for statistics of the 20 words, including polysemy), as one would expect from a classification task with a higher number of classes (near zero average correlation across settings). In the following we also discuss other relevant points with respect to Most Frequent Sense (MFS) bias and fine-tuning.

MFS Bias. As expected, macro-F1 results degrade for the purely supervised classification models (FastText and fine-tuned BERT), indicating the inherent sense biases captured by the model that lead to lowered performance for the obscure senses (see the work by Postma et al. (2016) for a more thorough analysis on this issue). However, BERT proves to be much more robust with this respect whereas FastText suffers heavily (highlighted in the macro setting).

Impact of Fine-Tuning. On average, fine-tuning improves the performance for BERT-Large by 1.6 points in terms of micro-F1 (from 95.8% to 97.5%) but decreases on macro-F1 (from 96.4% to 95.1%). While BERT-Base significantly correlates with BERT-Large in the 1NN setting (Pearson correlation above 0.9 for both micro and macro), it has a relatively low correlation with the fine-tuned BERT-Base (0.60 on micro-F1 and 0.75 on macro-F1). The same trend is observed for BERT-Large, where the correlation between fine-tuning and 1NN is 0.71 and 0.63 on micro-F1 and macro-F1, respectively. The operating principles behind both approaches are significantly different, which may explain this relatively low correlation. While fine-tuning is optimizing a loss function during training, the 1NN approach is simply memorizing states. By optimizing losses, fine-tuning is more susceptible to overfit on the MFS. In contrast, by memorizing states, 1NN models sense independently and disregard sense distributions entirely. These differences can explain the main discrepancies between the two strategies, reflected for both micro and macro scores (macro-F1 penalizes models that are not as good for less frequent senses). The differences between 1NN and fine-tuned models will be analyzed in more detail in our analysis section (Section 6).

In our error analysis we will show, among other things, that there are some cases that are difficult even for humans to disambiguate, for example, the intended meaning of *apple* (fruit vs. company) or *club* (nightclub vs. association) in the following contexts taken from the test set: "it also likes apple" and "she was discovered in a club by the record producer peter harris."

20 Given that the human performance is estimated based on a small subset of the test set, and given the skewed distribution of sense frequencies, macro-F1 values can be highly sensitive to less-frequent senses (which might even have no instance in the subset); hence, we do not report macro-F1 for human performance.

Computational Linguistics Volume 47, Number 2

5.3 Out of Domain

To verify the robustness of BERT and to see if the conclusions can be extended to other settings, we carried out a set of cross-domain evaluations in which the same BERT models (trained on CoarseWSD-20) were evaluated on the out-of-domain data set described in Section 4.4.

Table 6 shows the results. The performance trend is largely in line with that presented in Table 5, with some cases even having higher performance in this out-of-domain test set. Despite the relatively limited size of this test set, these results seem to corroborate previous findings and highlight the generalization capability of language models to perform WSD in different contexts. The fine-tuned version of BERT clearly achieves the highest micro-F1 scores, in line with previous experiments. Perhaps more surprisingly, BERT-Base 1NN achieves the best macro-F1 performance, also highlighting its competitiveness with respect to BERT-Large in this setting. As explained before, the 1NN strategy seems less prone to biases than the fine-tuned model, and this experiment shows the same conclusion extends to domain specificity as well, therefore the higher figures according to the macro metric. Interestingly, BERT-Base produces better results according to macro-F1 in the 1NN setting, despite lagging behind according to micro-F1. This suggests that data-intensive methods (e.g., fine-tuning) do not generally lead to significantly better results. Indeed, the results in Table 5 also confirm that the gains using a larger BERT model are not massive.

6. Analysis

In this section we perform an analysis on different aspects relevant to WSD on the CoarseWSD-20 data set. In particular, we first present a qualitative analysis on the type of contextualized embeddings learned by BERT (Section 6.1) and then analyze the impact of sense distribution of the training data (Section 6.2.1) as well as its size (Section 6.3) on WSD performance. Finally, we carry out an analysis on the inherent sense biases present in the pretrained BERT models (Section 6.4).

Table 6
Out-of-domain WSD results: Models trained on the CoarseWSD-20 training set and tested on the out-of-domain test set.

	Micro F1				**Macro F1**			
	1NN		**F-Tune**		**1NN**		**F-Tune**	
	BRT-B	BRT-L	BRT-B	BRT-L	BRT-B	BRT-L	BRT-B	BRT-L
bank	97.9	100.0	92.4	93.1	96.4	100.0	89.8	90.5
chair	100.0	100.0	98.3	99.2	100.0	100.0	94.8	97.4
pitcher	82.4	100.0	100.0	100.0	90.0	100.0	100.0	100.0
pound	89.1	87.0	96.4	94.9	94.0	81.5	85.5	77.5
spring	100.0	96.8	94.6	96.8	100.0	91.7	91.2	90.5
square	73.1	73.1	93.6	96.2	89.4	89.4	83.2	92.6
club	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
AVG	91.8	93.8	96.5	**97.2**	**95.7**	94.7	92.1	92.6

6.1 Contextualized Embeddings

The strong performance of the BERT-based 1NN WSD method reported for both fine and coarse-grained WSD proves that the representations produced by BERT are sufficiently precise to allow for effective disambiguation. Figures 2 and 3 illustrate the 2-D semantic space for contextualized representations of two target words (*square* and *spring*) in the test set. For each case, we applied the dimensionality technique that produced the most interpretable visualization, considering UMAP (McInnes et al. 2018) and Principal Component Analysis (PCA), although similar observations could be made using either of these two techniques. BERT is able to correctly distinguish and place most occurrences in distinct clusters. Few challenging exceptions exist, for example, two geometric senses of *square* are misclassified as public-square, highlighted in the figure ("... small *square* park located in ..." and " ... the narrator is a *square* ..."). Another interesting observation is for the season meaning of *spring*. BERT not only places all the contextualized representations for this sense in the same proximity in the space, it also makes a fine-grained distinction for the spring season of a specific year (e.g., "... in *spring* 2005 ...").

Beyond simply checking whether the nearest neighbor corresponds to the correct sense, there is still the question of the extent to which these representations are differentiated. In order to quantitatively analyze this, we plotted the distribution of cosine similarities between the contextual embeddings of the target word (to be disambiguated) from the test set and the closest predicted sense embedding learned from the training set. In Figure 4 we grouped these similarities by correct and incorrect predictions,

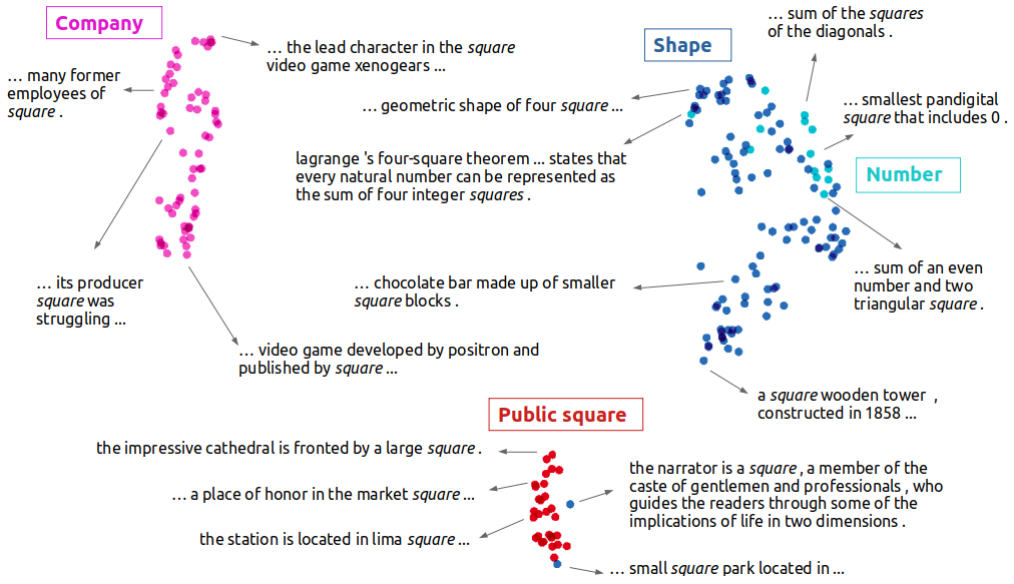


Figure 2
2-D visualizations of contextualized representations for different occurrences of *square* in the test set. While the company and public-square senses are grouped into distinct clusters, the numerical and geometrical meanings mostly overlap. Using UMAP for dimensionality reduction.

Computational Linguistics Volume 47, Number 2

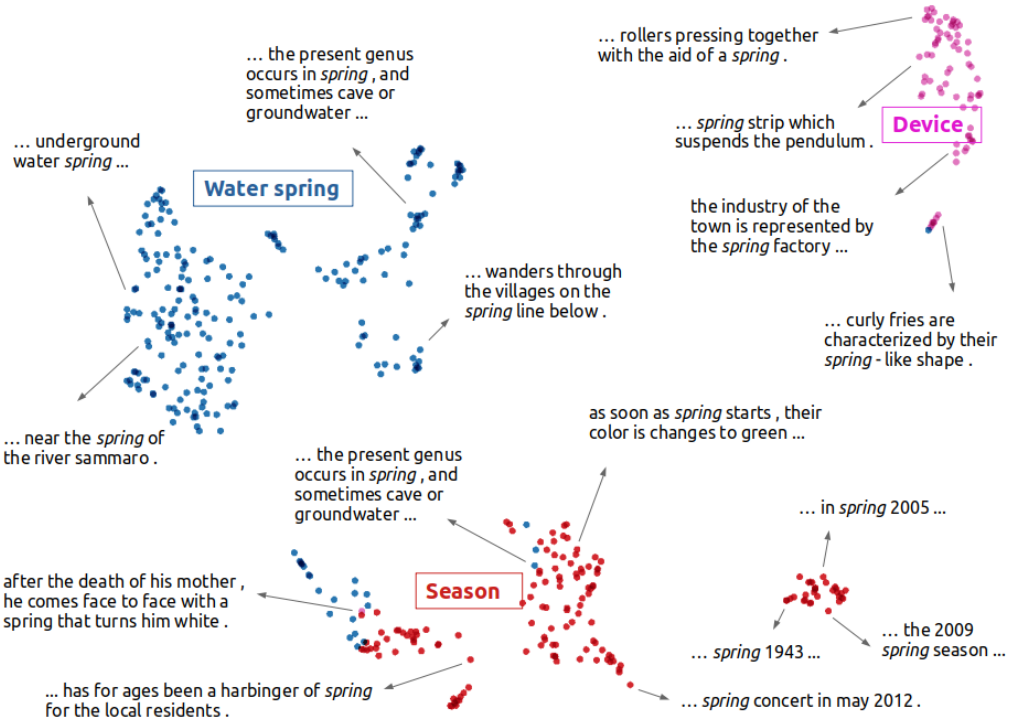


Figure 3
2-D visualizations of contextualized representations for different occurrences of *spring*. A fine-grained distinction can be observed for the season meaning of *spring*, with a distinct cluster (on the right) denoting the spring of a specific year. Using PCA for dimensionality reduction.

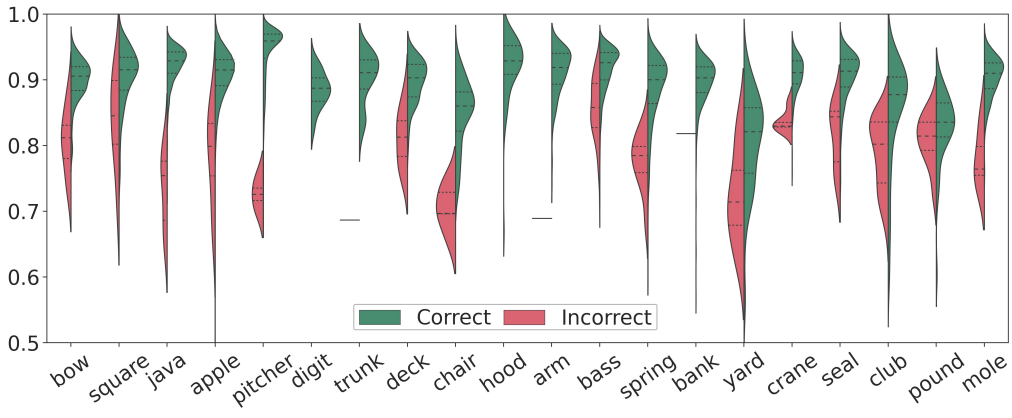


Figure 4
Distribution of cosine similarities between contextual embeddings (BERT-Large) of words to be disambiguated (in test set) and their corresponding closest sense embeddings learned from training data, for each word in the CoarseWSD-20 data set, grouped by correct and incorrect prediction.

revealing substantially different distributions. While incorrect prediction spans across the 0.5–0.9 interval, correct predictions are in the main higher than 0.75 for most words (over 97% of all predictions using BERT-Large with similarity higher than 0.75 are correct, for example). Consequently, this analysis also shows that a simple threshold could be used for effectively discarding false matches, increasing the precision of 1NN methods.

6.2 Role of Training Data

In order to gain insights on the role of training data, we perform two types of analysis: (1) distribution of training data—in particular, a comparison between skewed and balanced training sets (Section 6.2.1), and (2) the size of the training set (Section 6.2.2).

6.2.1 Distribution. To verify the impact of the distribution of the training data, we created a balanced training data set for each word by randomly removing instances for the more frequent senses in order to have a balanced distribution over all senses. Note that the original CoarseWSD-20 data set has a skewed sense distribution, given that it is constructed based on naturally occurring texts.

Table 7 shows the performance drop or increase when using a fully balanced training set instead of the original CoarseWSD-20 skewed training set (tested on the original skewed test set). Performance is generally similar across the two settings for the less entropic words (on top) that tend to have more uniform distributions. For the more entropic words (e.g., *deck*, *bank*, or *pitcher*), even though balancing the data

Table 7
Performance drop or increase when using a fully balanced training set instead of the original CoarseWSD-20 skewed training set.

	Micro F1						Macro F1					
	Static emb.		1NN		F-Tune		Static emb.		1NN		F-Tune	
	FTX-B	FTX-C	BRT-B	BRT-L	BRT-B	BRT-L	FTX-B	FTX-C	BRT-B	BRT-L	BRT-B	BRT-L
crane	−3.8	0.0	0.6	0.0	0.0	0.0	−3.7	0.0	0.6	0.0	0.0	0.0
java	−0.1	0.1	0.0	0.0	0.0	−0.1	−30.3	−15.1	0.1	0.0	0.0	−0.1
apple	−0.2	−0.6	0.0	0.0	0.0	−0.1	0.4	−0.4	0.0	0.0	0.0	−0.1
mole	−11.2	−1.5	0.0	0.0	−0.7	−0.7	−0.9	2.0	0.0	0.0	−0.5	−0.7
spring	−5.0	−2.0	0.0	0.2	−1.1	−0.9	−12.3	1.5	−0.2	0.1	−1.0	−0.7
chair	−6.2	−3.1	0.0	0.0	−1.0	0.3	−4.5	−2.3	0.0	0.0	−1.2	0.3
hood	−7.3	−1.2	0.0	−1.2	−0.4	0.0	12.2	4.4	−0.8	−1.5	−0.9	−0.3
seal	−23.1	−7.2	0.3	0.0	−2.9	−0.7	−9.0	−11.5	0.2	0.0	−7.3	−2.4
bow	−9.3	−3.7	0.0	0.0	−1.4	−0.8	−2.3	−2.0	0.0	0.0	−1.8	−1.5
club	−16.8	−5.9	0.0	−1.5	−0.8	−3.0	−8.6	−0.6	−0.3	−1.5	−0.4	−2.4
trunk	−13.0	−9.1	−3.9	0.0	−0.9	−1.7	−6.4	−4.3	−2.1	0.0	−0.9	−1.7
square	−23.7	−8.2	−6.8	−7.7	−4.7	−1.3	1.4	9.6	−3.4	−3.9	−4.8	1.1
bfarm	−2.4	−1.2	0.0	0.0	0.0	0.0	0.6	−0.8	0.0	0.0	0.0	0.0
digit	−16.7	−7.1	0.0	0.0	0.8	0.0	1.5	−4.5	0.0	0.0	1.2	0.0
bass	−9.1	−8.2	0.4	0.8	−5.1	−4.4	6.8	6.5	0.5	0.9	−5.6	−4.0
yard	−12.5	−5.6	−2.8	−4.2	−6.0	−2.3	18.2	11.6	−1.6	−2.5	−8.9	−3.9
pound	−34.0	−24.7	0.0	−1.0	−8.9	−1.4	18.5	36.7	7.5	−0.6	−8.8	2.0
deck	−26.3	−9.1	−2.0	−1.0	−5.7	−3.7	12.3	28.1	−1.1	−0.5	−5.0	2.1
bank	−17.4	−10.3	0.2	0.0	−2.6	−1.9	10.3	9.7	2.3	0.0	−10.6	−6.5
pitcher	−13.0	−6.4	−0.1	0.0	−1.3	−0.4	16.8	22.4	0.0	0.0	−26.7	−12.7
AVG	−12.6	−5.8	−0.7	−0.8	−2.1	−1.1	1.0	4.6	0.1	−0.5	−4.2	−1.6

Computational Linguistics Volume 47, Number 2

inevitably reduces the overall number of training instances to a large extent, it can result in improved macro results for FastText, and even improved macro-recall results for fine-tuning, as we will see in Table 8.

This can be attributed to the better encoding of the least frequent senses, which corroborates the findings of Postma, Izquierdo Bevia, and Vossen (2016) for conventional supervised WSD models, such as IMS or, in this case, FastText. In contrast, the micro-averaged results clearly depend on accurately knowing the original distribution in both the supervised and fine-tuning settings, as was also discussed in previous works (Bennett et al. 2016; Pasini and Navigli 2018). Moreover, the feature extraction procedure (1NN in this case) is much more robust to training distribution changes. Indeed, being solely based on vector similarities, the 1NN strategy is not directly influenced by the number of occurrences of each sense in the CoarseWSD-20 training set.

To complement these results, Table 8 shows the performance difference on the MFS (Most Frequent Class) and LFS (Least Frequent Class) classes when using the balanced training set. The most interesting takeaway from this experiment is the marked difference between precision and recall for the LFS in entropic words (bottom). While the recall of the BERT-Large fine-tuned model increases significantly (up to 52.4 points in the case of *deck*), the precision decreases (e.g., −27.1 points for *deck*). This means that the model is clearly less biased toward the MFS with a balanced training set, as we could expect. However, the precision for LFS is also lower, due to the model's lower sensitivity for higher-frequency senses. In general, these results suggest that the fine-tuned

Table 8
Precision and recall drop or increase on the Most Frequent Sense (MFS) and Least Frequent Sense (LFS) classes when using a fully balanced training set.

	F-Tune (BRT-L)				1NN (BRT-L)			
	Precision		Recall		Precision		Recall	
	MFS	LFS	MFS	LFS	MFS	LFS	MFS	LFS
crane	0.4	−0.4	−0.4	0.4	0.0	0.0	0.0	0.0
java	0.0	−0.3	−0.2	0.0	0.0	0.0	0.0	0.0
apple	−0.1	−0.1	−0.1	−0.2	0.0	0.0	0.0	0.0
mole	−0.9	−0.8	−0.9	−1.5	0.0	0.0	0.0	0.0
spring	−0.6	−1.0	−1.3	−1.4	0.0	0.6	0.0	0.0
chair	0.0	0.9	0.4	0.0	0.0	0.0	0.0	0.0
hood	0.7	0.0	0.0	−2.6	−2.1	0.0	0.0	0.0
seal	−0.3	0.0	−0.5	0.0	0.0	0.0	0.0	0.0
bow	0.8	−1.0	−0.6	−1.4	0.0	0.0	0.0	0.0
club	−3.4	−1.6	−2.2	−6.9	−3.9	0.0	0.9	−5.5
trunk	0.7	−7.4	−3.6	0.0	0.0	0.0	0.0	0.0
square	6.5	−0.5	−9.4	0.0	−0.4	0.0	−15.5	0.0
arm	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
digit	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
bass	2.5	−0.4	−8.6	−0.8	−0.7	1.8	0.7	1.1
yard	0.5	−14.0	−3.3	3.0	0.0	−7.9	−4.9	0.0
pound	4.0	−23.4	−5.8	36.7	−2.4	0.0	0.0	−1.1
deck	3.9	−27.1	−8.0	52.4	−2.9	0.0	0.0	−1.1
bank	0.8	−32.5	−2.8	15.2	0.0	0.0	0.0	0.0
pitcher	0.1	−46.0	−0.4	10.3	0.0	0.0	0.0	0.0
AVG	0.8	−7.8	−2.4	5.2	−0.6	−0.3	−0.9	−0.3

Table 9
Macro- and micro-F1 % performance for the two BERT-Large models. The last two rows indicate the F1 performance on the Most Frequent Sense (MFS) and Least Frequent Sense (LFS) classes.

	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	**5%**	**10%**	**25%**	**50%**	**ALL%**	**1%**	**5%**	**10%**	**25%**	**50%**	**ALL%**
Macro	74.2	81.6	85.8	91.5	94.2	95.1	94.4	95.3	95.6	95.8	96.0	96.4
Micro	89.0	93.5	95.3	96.3	97.0	97.5	95.5	95.8	95.7	95.7	95.6	95.8
MFS	91.9	95.3	96.4	97.2	97.5	98.0	95.8	95.8	95.6	95.6	95.4	95.4
LFS	52.1	64.3	71.9	83.4	88.5	91.0	91.6	93.3	94.1	94.6	95.5	96.6

BERT model is overly sensitive to the distribution of the training data, while its feature extraction counterpart suffers considerably less from this issue. In Section 6.4 we will extend the analysis on the bias present in each of the models.

6.2.2 Size. We performed an additional experiment to investigate the impact of training data size on the performance for the most and least frequent senses. To this end, we shrank the training data set for all words, while preserving their original distribution. Table 9 shows a summary of the aggregated micro-F1 and macro-F1 results, including the performance on the most and least frequent senses.[21] Clearly, the 1NN model performs considerably better than fine-tuning in settings with low training data (e.g., 74.2% to 94.4% macro-F1 with 1% of the training data). Interestingly, the 1NN's performance does not deteriorate with few training data, as the results with 1% and 100% of the training data do not vary much (less than two absolute points decrease in performance for micro-F1 and 0.3 in terms of micro-F1). Even for the LFS, the overall performance with 1% of the training data is above 90 (i.e., 91.6). This is an encouraging behavior, as in real settings sense-annotated data is generally scarce.

To obtain a more detailed picture for each word, Table 10 shows the macro-F1 results for each word and training size.[22] Again, we can observe a large drop for the most entropic words in the fine-tuning setting. Examples of words with a considerable degrading performance are *pitcher* or *bank*, which decrease from macro-F1 scores higher than 95% in both cases (97.3 and 95.6, respectively) to as low as 49.9 and 50.2 (almost random chance) with 1% of the training data, and still lower than 75% with 10% of the training data (63.9 and 74.9, respectively). This trend clearly highlights the need for gathering reasonable amounts of training data for the obscure senses. Moreover, this establishes a trade-off between balancing or preserving the original skewed distribution depending on the end goal, as discussed in Section 6.2.1.

6.3 *n*-Shot Learning

Given the results of the previous section, one may wonder how many instances would be enough for BERT to perform well in coarse-grained WSD. To verify this, we fine-tuned BERT on limited amounts of training data, with uniform distribution over word senses, each having between 1 (i.e., one-shot) and 30 instances. Figure 5 shows the

21 In the Appendix we include detailed results for each word and their MFS and LFS performance.
22 In the Appendix we include the same table for the micro-F1 results.

Computational Linguistics Volume 47, Number 2

Table 10
Macro-F1 results on the CoarseWSD-20 test set using training sets of different sizes sampled from the original training set.

	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	**5%**	**10%**	**25%**	**50%**	**ALL**	**1%**	**5%**	**10%**	**25%**	**50%**	**ALL**
crane	83.3	95.7	95.7	96.8	95.5	98.1	96.4	96.6	96.7	96.7	96.7	96.7
java	99.0	99.1	99.6	99.5	99.6	99.7	99.6	99.6	99.6	99.6	99.6	99.6
apple	99.3	99.4	99.4	99.4	99.5	99.6	99.1	99.1	99.1	99.1	99.1	99.1
mole	79.8	94.8	97.6	99.3	99.3	99.2	98.6	99.1	99.0	99.0	99.0	99.0
spring	94.8	97.6	96.8	96.9	97.8	98.1	97.9	97.9	97.9	97.9	97.9	97.8
chair	76.2	92.2	95.2	96.1	96.4	95.5	94.3	94.6	94.7	94.7	94.7	94.7
hood	57.2	89.3	92.3	96.6	97.7	99.6	94.7	98.6	99.2	99.5	100.0	100.0
seal	80.3	95.8	96.5	98.2	98.0	98.6	98.6	98.6	98.7	98.6	98.6	98.5
bow	49.3	86.8	95.7	96.0	97.5	98.6	93.5	96.0	96.2	95.9	95.7	95.7
club	70.1	77.4	77.0	80.0	83.0	84.1	85.6	86.5	87.4	87.6	88.0	88.7
trunk	77.9	84.6	97.5	98.6	98.6	98.0	97.7	98.3	98.7	99.3	99.3	99.3
square	68.4	69.6	73.5	76.6	79.4	91.4	86.7	88.0	87.8	88.1	91.1	94.7
arm	90.1	98.1	99.2	99.2	99.2	99.2	99.6	99.6	99.6	99.6	99.6	99.6
digit	92.4	79.7	92.1	98.8	100.0	100.0	99.1	100.0	100.0	100.0	100.0	100.0
bass	72.2	79.4	84.3	86.7	87.8	87.6	83.1	83.8	84.4	84.8	84.8	84.0
yard	82.7	85.7	88.3	94.3	99.1	99.1	93.4	93.4	92.8	92.6	92.2	93.4
pound	53.5	50.4	47.3	52.6	83.2	83.9	87.0	92.4	93.3	93.2	94.3	94.3
deck	56.7	48.2	48.2	70.2	77.2	78.0	85.5	85.1	88.9	91.1	92.1	95.7
bank	50.2	55.9	74.9	97.1	95.7	95.6	97.0	98.6	98.9	98.5	97.7	97.7
pitcher	49.9	52.3	63.9	96.5	99.3	97.3	100.0	100.0	100.0	100.0	100.0	100.0
Average	74.2	81.6	85.8	91.5	94.2	95.1	94.4	95.3	95.6	95.8	96.0	96.4

performance of both 1NN and fine-tuning strategies on this set of experiments. Perhaps surprisingly, we can see how having only three instances per sense is enough for achieving a competitive result. Then, only small improvements can be obtained by adding more instances. This is relevant in the context of WSD, as generally current sense-annotated corpora follow Zipf's law (Zipf 1949), and therefore contain many repeated senses that are very frequent. Significant improvements may therefore be obtained by simply getting a few sense annotations for less frequent instances. Figure 6 summarizes Figure 5 by showing the distribution of words according to their performance in the two strategies. In the case of fine-tuning, the performance is generally better in terms of micro compared with macro F-score. This further corroborates the previous observation, that there is a bias toward the most frequent sense (cf. Section 6.2.1). Additionally, in contrast to 1NN, fine-tuning greatly benefits from the increase in the training-data size, which also indicates the more robust behavior of 1NN strategy compared to its counterpart (cf. Section 6.2.1).

6.4 Bias Analysis

Supervised classifiers are known to have label bias toward more frequent classes, that is, those that are seen more frequently in the training data (Hardt et al. 2016), and this is particularly noticeable in WSD (Postma, Izquierdo Bevia, and Vossen 2016; Blevins and Zettlemoyer 2020). Label bias is a reasonable choice for maximizing performance when the distribution of classes is skewed, particularly for classification tasks with a small number of categories (which is often the case in WSD). For the same reason, many

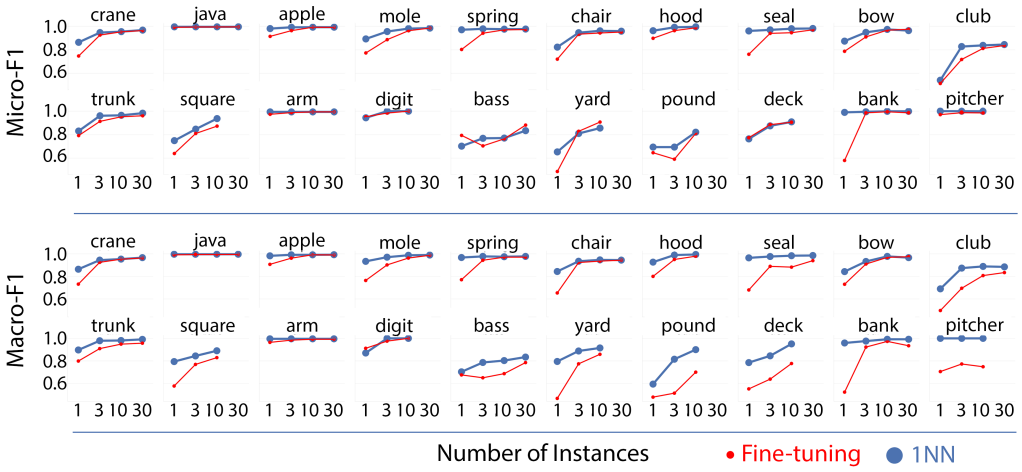


Figure 5
Micro and macro F-scores for different values of n in the n-shot setting, for all the words and for the two WSD strategies. Results are averaged from three runs over three different samples.

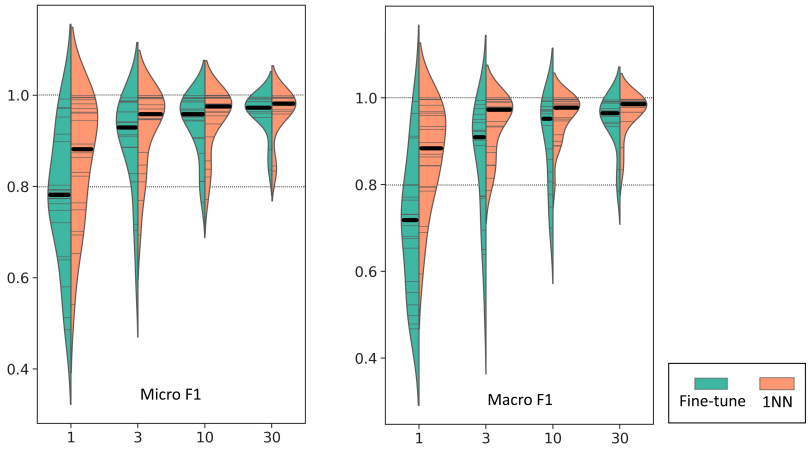


Figure 6
Distribution of performance scores for all 20 words according to micro and macro F1 in the two WSD strategies (left: fine-tuning, right: 1NN) and for different values of n—i.e., 1, 3, 10, 30 (if available).

of the knowledge-based systems are coupled with the MFS back-off strategy: When the system is not confident in its disambiguation, it backs off to the most frequent sense (MFS) of the word (instead of resorting to the low-confidence decision).

We were interested in investigating the inherent sense biases in the two BERT-based WSD strategies. We opted for the n-shot setting given that it provides a suitable setting for evaluating the relationship between sense bias and training data size. Moreover, given that the training data in the n-shot setting is uniformly distributed (balanced), the impact of sense-annotated training data in introducing sense bias is minimized. This analysis is mainly focused on two questions: (1) how do the two strategies (fine-tuning

Computational Linguistics Volume 47, Number 2

Table 11
Average sense bias values (B) for the two WSD strategies and for different values of n.

One-shot		3-shot		10-shot		30-shot	
F-Tune	1NN	F-Tune	1NN	F-Tune	1NN	F-Tune	1NN
0.232	0.137	0.111	0.078	0.050	0.052	0.021	0.025

and 1NN) compare in terms of sense bias?, and (2) what are the inherent sense biases (if any) in the pretrained BERT language model?

6.4.1 Sense Bias Definition. We propose the following procedure for computing the disambiguation bias toward a specific sense.[23] For a word with polysemy n, we are interested in computing the disambiguation bias B_j toward its j^{th} sense (s_j). Let n_{ij} be the total number of test instances with the gold label s_i that were mistakenly disambiguated as s_j ($i \neq j$). We first normalize n_{ij} by the total number of (gold-labeled) instances for s_i, that is, $\Sigma_j n_{ij}$, to obtain bias b_{ij}, which is the bias from sense i to sense j. In other words, b_{ij} denotes the ratio of s_i-labeled instances that were misclassified as s_j. The total bias toward a specific sense, B_j, is then computed as:

$$B_j = \sum_{\substack{i=1 \\ i \neq j}}^{n} \left(\frac{n_{ij}}{\Sigma_j n_{ij}}\right) \tag{4}$$

The value of B_j denotes the tendency of the disambiguation system to disambiguate a word with the intended sense of s_k, $k \neq j$, incorrectly as s_j. The higher the value of B_j, the more the disambiguation model is biased towards s_j. We finally compute the **sense bias** B as the *maximum* B_j value toward different senses of a specific word, that is, $max(B_j), j \in [1, n]$. Given fluctuations in the results, particularly for the case of small training data, we take the median of three runs to compute B_j.

In our coarse-grained disambiguation setting, the bias B can be mostly attributed to the case where the system did not have enough evidence to distinguish s_j from other senses and had pretraining bias towards s_j. One intuitive explanation for this would be that the language model is biased toward s_j because it has seen the target word more often with this intended sense than other $s_{k,j \neq k}$ senses.

6.4.2 Results. Table 11 reports the average sense bias values (B) for the two WSD strategies and for different values of n (training data size) in the n-shot setting. We also illustrate using radar charts in Figure 7 the sense bias for a few representative cases. The numbers reported in the figure (in parentheses) represent the bias value B for the corresponding setting (word, WSD strategy, and n's value).

Based on our observations, we draw the following general conclusions.

Bias and Training Size. There is a consistent pattern across all words and for both the strategies: Sense bias rapidly reduces with increase in the training data. Specifically, the

23 The procedure can presumably be used for quantifying bias in other similar classification settings.

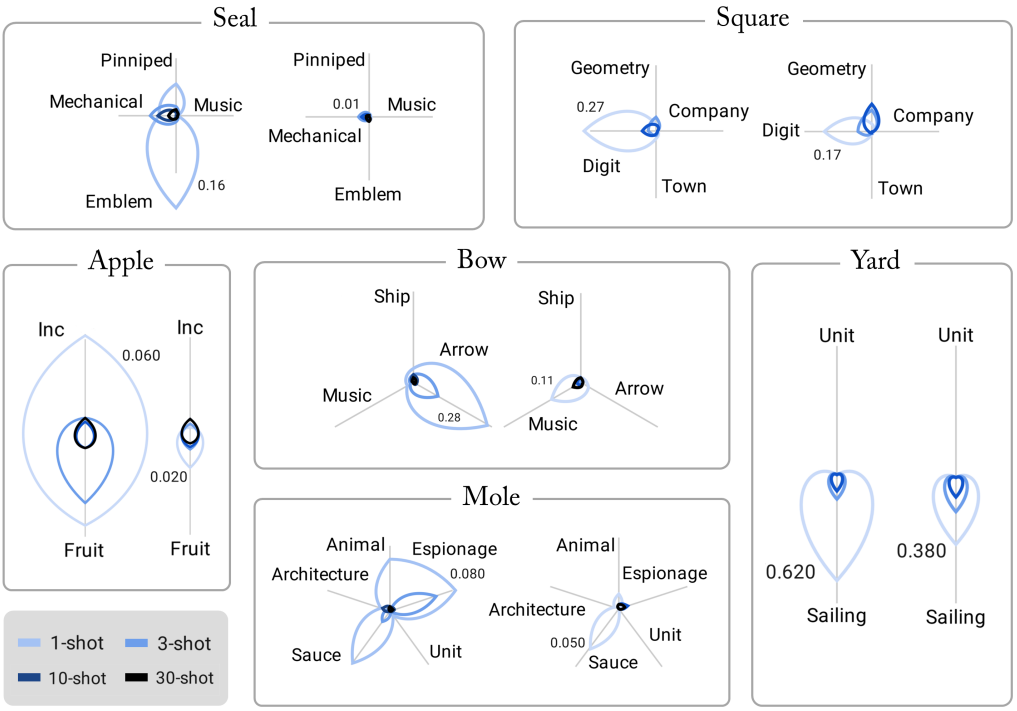


Figure 7
Sense bias for a few representative cases from each polysemy class for the two WSD strategies (left: fine-tuning, right: 1NN) and for different values of n, i.e., 1, 3, 10, 30 (if available).

average bias B approximately reduces by half with each step of increase in the training size. This is supported by the radar charts in Figure 7 (see, for instance, *apple*, *yard*, and *bow*). The WSD system tends to be heavily biased in the one-shot setting (particularly in the fine-tuning setting), but the bias often improves significantly with just 3 instances in the training data (3-shot).

Disambiguation Strategy: 1NN vs. Fine-Tuning. Among the two WSD strategies, the 1NN approach proves to be more robust with respect to sense biases. This is particularly highlighted in the one-shot setting where the average sense bias value is 0.137 for 1NN in comparison to 0.232 for fine-tuning. The trend is also clearly visible for almost all words in the radar charts in Figure 7. This corroborates our findings in Section 6.3 that the 1NN strategy is the preferable choice particularly with limited data. For higher values of n (larger training sizes) the difference between the two strategies diminishes, with both settings proving robust with respect to sense bias.

It is also notable that the two strategies, despite being usually similar in behavior, might not necessarily have matching biases toward the same senses. For instance, the fine-tuning setting shows bias only toward the arrow sense of *bow*, whereas 1NN is instead (slightly) biased toward its music sense. Another example is for the word *digit* for which with the same set of training instances in the one-shot setting (one sentence for each of the two senses), all the mistakes (5 in total) of the fine-tuning model are

numerical digits incorrectly tagged as anatomical, whereas all the mistakes in the 1NN setting (5 in total) are the reverse.

Finally, we also observed that for cases with subtle disambiguation, both the strategies failed consistently in the one-shot setting. For instance, a common mistake shared by the two strategies was for cases where the context contained semantic cues for multiple senses, for example, "the English word *digit* as well as its translation in many languages is also the anatomical term for fingers and toes." in which the intended meaning of *digit* is the numerical one (both strategies failed on disambiguation for this). This observation is in line with the analysis of Reif et al. (2019), which highlighted the failure of BERT in identifying semantic boundaries of words.

Pretraining Label Bias. In most of the conventional supervised WSD classifiers (such as IMS), which rely on sense-annotated training data as their main source of information, the source of sense bias is usually the skewed distribution of instances for different senses of a word (Pilehvar and Navigli 2014). For instance, the word *digit* would appear much more frequently with its numerical meaning than the finger meaning in an open-domain text. Therefore, a sense-annotated corpus that is sampled from open-domain texts shows a similar sense distribution, resulting in a bias toward more frequent senses in the classification.

Given that in the *n*-shot setting we restrict the training data sets to have a uniform distribution of instances, sense bias in this scenario can be indicative of inherent sense biases in BERT's pretraining. We observed that the pretrained BERT indeed exhibits sense biases, often consistently across the two WSD strategies. For instance, we observed the following biases toward (often) more frequent senses of words: *java* toward its programming sense (rather than island), *deck* toward ship deck (rather than building deck), *yard* toward its sailing meaning (rather than measure unit), and *digit* and *square* toward their numerical meanings. We also observed some contextual cues that misled the WSD system, especially in the one-shot setting. For instance, we observed that our BERT-based WSD system had a tendency to classify *square* as its digit meaning whenever there was a *number* in its context, for example, "marafor is a roman square with two temples attached" or "it has 4 trapezoid and 2 square faces." Not surprisingly, the source of most bias toward the digit sense of *square* is from its geometrical sense (which has domain relatedness). Also, classification for *digit* was often biased toward its numerical meaning. Similarly to the case of *square*, the existence of a number in context seems to bias the model toward numerical meanings, for example, "There were five *digit* on each hand and four on each foot."

Sensitivity to Initialization. We observed a high variation in the results, especially for the one-shot setting, suggesting the high sensitivity of the model with little evidence from training to the initialization point. For instance, in the one-shot experiment for the fine-tuning model and the word *bank*, in three runs, 1%, 60%, and 70% of the test instances for the financial bank are incorrectly classified as river bank. Similarly, for *crane*, 12%, 25%, and 72% of the machine instances are misclassified as bird in three runs. The 1NN strategy, in addition to being less prone to sense biases, is generally more robust across multiple runs. For these two examples, the figures are 2%, 0%, and 0% for *bank* and 15%, 0%, and 27% for *crane*. Other than the extent of bias, we observed that the direction can also change dramatically from run to run. For example, in the one-shot 1NN setting and for the word *apple*, almost all the mistakes in the first two runs (37 of 38 and 12 of 14) were incorporation for fruit, whereas in the third run, almost all (6 of 7) were fruit for incorporation.

7. Discussion

In the previous sections we have run an extensive set of experiments to investigate various properties of language models when adapted to the task of WSD. In the following we discuss some of the general conclusions and open questions arising from our analysis.

Fine-Grained vs. Coarse-Grained. A well-known issue of WordNet is the fine granularity of its sense distinctions (Navigli 2009). For example, the noun *star* has 8 senses in WordNet, two of which refer to a "celestial body," only differing in if they are visible from the Earth or not. Both meanings translate to *estrella* in Spanish and therefore this sense distinction serves no advantage in MT, for example. In fact, it has been shown that coarse-grained distinctions are generally more suited to downstream applications (Rüd et al. 2011; Severyn, Nicosia, and Moschitti 2013; Flekova and Gurevych 2016; Pilehvar et al. 2017). However, the coarsening of sense inventories is certainly not a solved task. Whereas in this article we relied either on experts for selecting senses from Wikipedia (given the reduced number of selected words) or domain labels from lexical resources for WordNet (Lacerra et al. 2020), there are other strategies for coarsening sense inventories (McCarthy, Apidianaki, and Erk 2016; Hauer and Kondrak 2020)—for instance, based on translations or parallel corpora (Resnik and Yarowsky 1999; Apidianaki 2008; Bansal, DeNero, and Lin 2012). This is generally an open problem, especially for verbs (Peterson and Palmer 2018), which have not been analyzed in-depth in this article due to lack of effective techniques for an interpretable coarsening. Indeed, while in this work we have shown how contextualized embeddings encode meaning to a similar extent as humans do, for fine-grained distinctions these have been shown to correlate to a much lesser extent, an area that requires further exploration (Haber and Poesio 2020).

Fine-Tuning vs. Feature Extraction (1NN). The distinction between fine-tuning and feature extraction has been already studied in the literature for different tasks (Peters, Ruder, and Smith 2019). The general assumption is that fine-tuned models perform better when reasonable amounts of training data are available. In the case of WSD, however, feature extraction (specifically the 1NN strategy explained in this article) is the more solid choice on general grounds, even when training data is available. The advantages of feature extraction (1NN) with respect to fine-tuning are 3-fold:

1. It is significantly less expensive to train as it simply relies on extracting contextualized embeddings from the training data. This is especially relevant when the WSD model is to be used in an all-words setting.
2. It is more robust to changes in the training distribution (see Section 6.2.1).
3. It works reasonably well for limited amounts of training data (see Section 6.2.2), even in few-shot settings (see Section 6.3).

Few-Shot Learning. An important limitation of supervised WSD models is their dependence on sense-annotated corpora, which is expensive to construct, that is, the so-called knowledge-acquisition bottleneck (Gale, Church, and Yarowsky 1992b; Pasini 2020). Therefore, being able to learn from a limited set of examples is a desirable property of WSD models. Encouragingly, as mentioned above, the simple 1NN method studied in this article shows robust results even with as few as three training examples per word sense. In the future it would be interesting to investigate models relying on knowledge

from lexical resources that can perform WSD with no training instances available (i.e., zero-shot), in the line of Kumar et al. (2019) and Blevins and Zettlemoyer (2020).

8. Conclusions

In this article we have provided an extensive analysis on how pretrained language models (particularly BERT) capture lexical ambiguity. Our aim was to inspect the capability of BERT in predicting different usages of the same word depending on its context, similarly as humans do (Rodd 2020). The general conclusion we draw is that in the ideal setting of having access to enough amounts of training data and computing power, BERT can approach human-level performance for coarse-grained noun WSD, even in cross-domain scenarios. However, this ideal setting rarely occurs in practice, and challenges remain to make these models more efficient and less reliant on sense-annotated data. As an encouraging finding, feature extraction-based models (referred to as 1NN throughout the article) show strong performance even with a handful of examples per word sense. As future work it would be interesting to focus on the internal representation of the Transformer architecture by, for example, carrying out an in-depth study of layer distribution (Tenney, Das, and Pavlick 2019), investigating the importance of each attention head (Clark et al. 2019), or analyzing the differences for modeling concepts, entities, and other categories of words (e.g., verbs). Moreover, our analysis could be extended to additional Transformer-based models, such as RoBERTa (Liu et al. 2019b) and T5 (Raffel et al. 2020).

To enable further analysis of this type, another contribution of the article is the release of the CoarseWSD-20 data set (Section 4), which also includes the out-of-domain test set (Section 4.4). This data set can be reliably used for quantitative and qualitative analyses in coarse-grained WSD, as we performed. We hope that future research in WSD will take inspiration on the types of analyses performed in this work, as they help shed light on the advantages and limitations of each approach. In particular, few-shot and bias analysis along with training distribution variations are key aspects to understanding the versatility and robustness of any given approach.

Finally, WSD is clearly not a solved problem, even in the coarse-grained setting, due to a few challenges: (1) it is an arduous process to manually create high-quality full-coverage training data; therefore, future research should also focus on reliable ways of automating this process (Taghipour and Ng 2015; Delli Bovi et al. 2017; Scarlini, Pasini, and Navigli 2019; Pasini and Navigli 2020; Loureiro and Camacho-Collados 2020; Scarlini, Pasini, and Navigli 2020b) and/or leveraging specific knowledge from lexical resources (Luo et al. 2018; Kumar et al. 2019; Huang et al. 2019); and (2) the existing sense-coarsening approaches are mainly targeted at nouns, and verb sense modeling remains an important open research challenge.

APPENDIX

Word-in-Context Evaluation

Word-in-Context (Pilehvar and Camacho-Collados 2019, WiC) is a binary classification task from the SuperGLUE language understanding benchmark (Wang et al. 2019) aimed at testing the ability of models to distinguish between different senses of the same word without relying on a predefined sense inventory. In particular, given a target word (either a verb or a noun) and two contexts where such target word occurs, the

Table 12
Sample positive (T) and negative (F) pairs from the WiC data set (target word in *italics*).

F	There's a lot of trash on the *bed* of the river | I keep a glass of water next to my *bed* when I sleep
F	*Justify* the margins | The end *justifies* the means
T	*Air* pollution | Open a window and let in some *air*
T	The expanded *window* will give us time to catch the thieves | You have a two-hour *window* of clear weather to finish working on the lawn

Table 13
Accuracy (%) performance of different models on the WiC data set.

Type	Model	Accuracy
Hybrid	KnowBERT (Peters et al. 2019)	70.9
	SenseBERT (Levine et al. 2020)	72.1
	LMMS-LR (Loureiro and Jorge 2019b)	68.1
Fine-tuned/Supervised	BERT-Base	69.6
	BERT-Large	69.6
	FastText-B	52.3
	FastText-C	54.7
Lowerbound	*Most Frequent Class*	50.0
Upperbound	*Human performance*	80.0

task consists of deciding whether the two target words in context refer to the same sense or not. Even though no sense inventory is explicitly given, this data set was also constructed based on WordNet. Table 12 shows a few examples from the data set.

BERT-Based Model. Given that the task in WiC is a binary classification, the 1NN model is not applicable because a training to learn sense margins is necessary. Therefore, we experimented with the BERT model fine-tuned on WiC's training data. We followed Wang et al. (2019) and fused the two sentences and fed them as input to BERT. A classifier was then trained on the concatenation of the resulting BERT contextual embeddings.

Baselines. In addition to our BERT-based model, we include results for two FastText supervised classifiers (Joulin et al. 2017) as baselines: a basic one with random initialization (FastText-B) and another initialized with FastText embeddings trained on the Common Crawl (FastText-C). As other indicative reference points, we added two language models that are enriched with WordNet (Levine et al. 2020; Loureiro and Jorge 2019b) and another with WordNet and Wikipedia (Peters et al. 2019).

Results. Table 13 shows the result of BERT models and the other baselines on the WiC benchmark.[24] We can see that BERT significantly outperforms the FastText static word embedding. The two versions of BERT (Base and Large) perform equally well on this task, achieving results close to the state of the art. As with fine-grained all-words WSD,

24 Data and results from comparison systems taken from `https://pilehvar.github.io/wic/`.

the additional knowledge drawn from WordNet proves to be beneficial, as shown by the results for KnowBERT and SenseBERT.

CoarseWSD-20: Sense Information

Table 17 shows for each sense their ID (as per their Wikipedia page title), definition, and example usage from the data set.

Complementary Results in CoarseWSD-20

1. Table 14 shows micro-F1 results for the experiment with different training data sizes sampled from the original CoarseWSD-20 training set (cf. Section 6.2.2 of the article).
2. Table 15 shows the micro-F1 performance for fine-tuning and 1NN and for varying sizes of the training data (with similar skewed distributions) for both Most Frequent Sense (MFS) and Least Frequent Sense (LFS) classes (cf. Section 6.2.2 of the article).
3. Table 16 includes the complete results for the n-shot experiment, including the FastText baselines (cf. Section 6.3 of the article).

Table 14
Micro-F1 results on the CoarseWSD-20 test set using training sets of different sizes sampled from the original training set.

	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	**5%**	**10%**	**25%**	**50%**	**ALL**	**1%**	**5%**	**10%**	**25%**	**50%**	**ALL**
crane	84.1	95.8	95.8	96.8	95.5	98.1	96.5	96.7	96.8	96.8	96.8	96.8
java	99.1	99.1	99.6	99.5	99.6	99.7	99.6	99.6	99.6	99.6	99.6	99.6
apple	99.4	99.4	99.5	99.5	99.5	99.6	99.2	99.2	99.2	99.2	99.2	99.2
mole	80.1	96.0	97.7	99.0	99.0	98.9	97.7	98.6	98.5	98.5	98.5	98.5
spring	95.0	97.5	96.9	96.8	97.8	98.3	98.0	98.0	98.0	98.0	97.9	97.8
chair	82.8	93.6	95.9	96.7	96.9	96.2	95.1	95.8	96.2	96.2	96.2	96.2
hood	77.6	90.7	93.5	97.2	97.6	99.6	97.2	99.0	99.4	99.6	100.0	100.0
seal	92.4	97.6	98.1	98.8	98.5	99.0	98.1	98.2	98.3	98.2	98.2	98.1
bow	74.1	92.4	96.1	96.7	97.5	98.5	94.9	95.9	95.8	95.5	95.3	95.3
club	72.8	78.7	78.7	80.4	83.5	84.7	82.0	82.9	83.8	84.0	84.4	85.1
trunk	86.2	88.7	97.8	98.7	98.7	98.3	97.8	98.2	98.4	98.7	98.7	98.7
square	88.4	87.3	92.6	92.4	92.9	95.7	93.9	94.2	94.1	95.2	95.7	96.1
arm	93.1	98.6	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4
digit	95.2	89.7	95.2	99.2	100.0	100.0	99.6	100.0	100.0	100.0	100.0	100.0
bass	92.0	93.4	94.4	95.1	95.6	95.8	86.6	86.1	85.8	85.5	85.2	84.5
yard	90.7	94.0	95.4	97.2	99.5	99.5	89.8	88.9	87.8	87.5	86.8	88.9
pound	88.3	90.0	89.7	89.0	94.9	94.9	92.6	92.8	92.0	90.4	89.7	89.7
deck	93.6	92.9	92.9	93.9	95.0	95.3	91.4	91.9	91.7	91.6	91.4	91.9
bank	95.2	95.5	97.1	99.5	99.3	99.3	99.7	99.9	99.9	99.9	99.8	99.8
pitcher	99.5	99.6	99.6	99.9	100.0	100.0	100.0	100.0	99.9	100.0	99.9	99.9
Average	89.0	93.5	95.3	96.3	97.0	97.5	95.5	95.8	95.7	95.7	95.6	95.8

Table 15
Micro-F1 performance for the two WSD strategies and for varying sizes of the training data (with similar skewed distributions) for the MFS (top) and LFS (bottom).

Most Frequent Sense (MFS)

	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	**5%**	**10%**	**25%**	**50%**	**ALL**	**1%**	**5%**	**10%**	**25%**	**50%**	**ALL**
crane	86.9	96.1	96.0	97.0	95.9	98.2	100.0	100.0	100.0	100.0	100.0	100.0
java	99.2	99.3	99.7	99.6	99.7	99.8	99.4	99.4	99.4	99.4	99.4	99.4
apple	99.5	99.5	99.6	99.6	99.6	99.7	99.5	99.5	99.5	99.5	99.5	99.5
mole	75.4	96.2	97.1	98.7	98.7	98.5	95.2	97.7	97.4	97.4	97.4	97.4
spring	96.0	97.7	97.2	97.0	98.0	98.8	97.7	97.8	97.8	97.7	97.7	97.5
chair	88.8	95.5	97.0	97.6	97.8	97.2	96.6	98.2	98.9	98.9	98.9	98.9
hood	91.6	93.4	95.6	98.3	97.9	99.7	100.0	100.0	100.0	100.0	100.0	100.0
seal	90.9	97.0	97.5	98.7	98.4	98.9	98.6	98.5	98.5	98.2	98.5	98.5
bow	85.5	97.7	97.2	98.2	98.2	98.7	97.6	98.1	98.3	98.3	98.3	98.3
club	74.6	80.4	80.6	81.9	84.1	85.2	79.5	78.5	78.5	78.4	78.2	77.8
trunk	90.6	91.4	98.2	98.9	98.9	98.6	97.9	97.9	97.9	97.9	97.9	97.9
square	89.6	88.5	93.0	92.8	93.2	95.7	93.7	93.6	93.4	95.8	95.1	94.2
arm	95.5	99.1	99.6	99.6	99.6	99.6	99.2	99.2	99.2	99.2	99.2	99.2
digit	97.0	93.9	97.1	99.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
bass	95.2	96.0	96.8	96.7	97.1	97.2	86.0	85.2	84.6	84.1	83.7	82.9
yard	94.4	96.6	97.4	98.4	99.7	99.7	88.3	86.9	85.7	85.2	84.4	86.9
pound	93.7	94.7	94.6	94.1	97.2	97.2	94.1	92.9	91.7	89.7	88.5	88.5
deck	96.7	96.3	96.3	96.8	97.3	97.5	92.4	93.0	92.1	91.7	91.3	91.3
bank	97.6	97.7	98.5	99.7	99.6	99.6	100.0	100.0	100.0	100.0	100.0	100.0
pitcher	99.8	99.8	99.8	100.0	100.0	100.0	100.0	100.0	99.9	100.0	99.9	99.9
Average	91.9	95.3	96.4	97.2	97.5	98.0	95.8	95.8	95.6	95.6	95.4	95.4

Least Frequent Sense (LFS)

	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	**5%**	**10%**	**25%**	**50%**	**ALL**	**1%**	**5%**	**10%**	**25%**	**50%**	**ALL**
crane	79.7	95.4	95.4	96.6	95.2	98.0	92.8	93.2	93.4	93.4	93.4	93.4
java	98.8	98.8	99.5	99.4	99.5	99.6	99.9	99.9	99.9	99.9	99.9	99.9
apple	99.2	99.2	99.3	99.3	99.4	99.5	98.7	98.7	98.7	98.7	98.7	98.7
mole	72.5	86.7	97.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
spring	95.0	98.2	97.0	97.9	97.9	97.5	97.3	97.3	97.3	97.3	97.3	97.3
chair	63.7	88.9	93.4	94.6	95.0	93.8	92.1	91.0	90.5	90.5	90.5	90.5
hood	65.7	82.5	89.2	95.2	95.2	99.2	97.0	97.3	97.7	98.5	100.0	100.0
seal	39.1	89.3	91.0	96.0	96.0	97.3	100.0	100.0	100.0	100.0	100.0	100.0
bow	0.0	73.2	95.3	94.6	98.0	99.4	91.0	98.5	100.0	100.0	100.0	100.0
club	63.5	74.4	73.3	80.0	81.6	82.5	95.2	95.2	95.2	95.2	95.2	95.2
trunk	49.7	66.6	95.2	100.0	100.0	96.5	95.2	97.1	98.2	100.0	100.0	100.0
square	9.5	21.4	4.8	20.5	30.3	76.0	53.8	58.5	57.7	56.4	69.2	84.6
arm	84.7	97.2	98.9	98.9	98.9	98.9	100.0	100.0	100.0	100.0	100.0	100.0
digit	87.7	65.5	87.2	98.0	100.0	100.0	98.1	100.0	100.0	100.0	100.0	100.0
bass	29.5	48.2	61.8	65.6	68.5	67.3	69.7	73.2	75.6	77.7	78.4	77.3
yard	70.9	74.8	79.2	90.3	98.4	98.4	98.5	100.0	100.0	100.0	100.0	100.0
pound	13.3	6.1	0.0	11.1	69.3	70.6	80.0	92.0	95.0	96.7	100.0	100.0
deck	16.7	0.0	0.0	43.6	57.1	58.6	78.6	77.1	85.7	90.5	92.9	100.0
bank	2.9	14.0	51.4	94.4	91.8	91.6	93.9	97.3	97.7	97.0	95.5	95.5
pitcher	0.0	4.8	28.0	93.0	98.7	94.6	100.0	100.0	100.0	100.0	100.0	100.0
Average	52.1	64.3	71.9	83.4	88.5	91.0	91.6	93.3	94.1	94.6	95.5	96.6

Table 16
Micro- and macro-F1 results in the *n*-shot setting for all the two BERT-based WSD strategies (as well as for the static embedding baseline) in our experiments and for all the words in the data set. Results are the average of three runs (standard deviation is shown in parentheses).

			Micro F1				**Macro F1**			
			1	3	10	30	1	3	10	30
crane	Static emb.	Fasttext-B	48.6 (5.3)	57.5 (5.3)	57.7 (3.5)	70.9 (5.2)	48.1 (4.5)	57.9 (4.8)	58.5 (3.3)	71.1 (5.3)
		Fasttext-C	52.7 (1.2)	69.4 (6.3)	82.0 (3.0)	83.4 (6.6)	51.4 (1.1)	69.6 (6.3)	81.9 (3.0)	83.5 (6.8)
	1NN	BERT-Base	84.5 (9.8)	93.8 (1.8)	93.6 (1.4)	94.5 (0.3)	84.3 (10.1)	93.7 (1.9)	93.4 (1.4)	94.4 (0.3)
		BERT-Large	86.4 (3.8)	94.7 (3.5)	95.5 (1.4)	96.8 (0.9)	86.4 (3.9)	94.5 (3.7)	95.4 (1.4)	96.7 (0.9)
	Fine-Tuning	BERT-Base	65.6 (1.9)	88.7 (7.8)	94.1 (3.2)	95.8 (0.6)	63.4 (1.3)	88.7 (7.8)	94.0 (3.3)	95.7 (0.6)
		BERT-Large	74.7 (10.5)	92.6 (2.2)	95.3 (1.7)	96.4 (1.3)	73.2 (12.3)	92.5 (2.2)	95.3 (1.7)	96.4 (1.3)
java	Static emb.	Fasttext-B	63.1 (1.3)	66.8 (2.4)	68.5 (2.8)	80.6 (6.8)	55.8 (3.5)	60.5 (3.9)	66.9 (1.6)	80.9 (6.4)
		Fasttext-C	78.4 (5.8)	90.1 (3.3)	90.9 (1.9)	94.9 (2.3)	78.1 (7.8)	90.3 (2.5)	90.5 (2.3)	95.1 (2.1)
	1NN	BERT-Base	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.7 (0.0)	99.7 (0.0)
		BERT-Large	99.6 (0.1)	99.6 (0.1)	99.6 (0.0)	99.6 (0.0)	99.7 (0.0)	99.7 (0.1)	99.7 (0.0)	99.7 (0.0)
	Fine-Tuning	BERT-Base	99.2 (0.4)	98.8 (0.7)	99.4 (0.2)	99.3 (0.1)	99.1 (0.5)	98.8 (0.7)	99.4 (0.2)	99.3 (0.1)
		BERT-Large	99.2 (0.6)	99.4 (0.1)	99.5 (0.1)	99.6 (0.1)	99.1 (0.6)	99.4 (0.1)	99.5 (0.1)	99.5 (0.1)
apple	Static emb.	Fasttext-B	43.1 (2.1)	52.0 (4.9)	55.2 (8.4)	74.7 (1.2)	45.4 (3.8)	55.3 (5.1)	61.3 (5.4)	73.9 (2.4)
		Fasttext-C	71.2 (9.9)	81.2 (2.9)	87.3 (2.1)	93.2 (0.2)	63.7 (12.8)	80.7 (1.8)	86.7 (3.0)	92.4 (0.3)
	1NN	BERT-Base	95.5 (3.9)	99.0 (0.1)	99.0 (0.1)	99.0 (0.0)	96.1 (3.2)	99.0 (0.1)	99.0 (0.1)	99.0 (0.0)
		BERT-Large	98.1 (1.2)	99.2 (0.0)	99.3 (0.1)	99.3 (0.1)	98.3 (0.9)	99.2 (0.1)	99.2 (0.1)	99.3 (0.1)
	Fine-Tuning	BERT-Base	90.7 (9.0)	98.3 (0.6)	99.0 (0.1)	99.0 (0.1)	89.6 (10.3)	98.2 (0.7)	98.9 (0.1)	98.9 (0.1)
		BERT-Large	91.5 (5.5)	96.4 (2.5)	99.3 (0.1)	99.1 (0.5)	90.7 (6.2)	96.2 (2.6)	99.2 (0.1)	99.0 (0.5)
mole	Static emb.	Fasttext-B	21.2 (9.9)	16.3 (7.3)	38.2 (1.1)	65.9 (1.6)	17.9 (2.2)	22.8 (3.9)	41.5 (9.2)	68.8 (2.0)
		Fasttext-C	48.7 (2.6)	63.3 (4.3)	75.9 (6.3)	88.0 (0.9)	57.3 (1.3)	68.6 (2.5)	79.4 (4.3)	89.4 (1.7)
	1NN	BERT-Base	75.9 (5.5)	91.1 (4.0)	95.1 (2.2)	97.4 (0.6)	84.9 (4.6)	93.9 (1.8)	96.6 (1.2)	97.7 (0.3)
		BERT-Large	89.3 (1.1)	95.6 (0.8)	98.1 (0.8)	98.5 (0.0)	93.4 (0.6)	97.1 (0.7)	98.8 (0.4)	99.0 (0.0)
	Fine-Tuning	BERT-Base	71.2 (4.1)	86.2 (5.2)	95.8 (1.3)	97.6 (0.4)	70.7 (4.8)	87.8 (3.3)	95.8 (1.4)	97.5 (0.6)
		BERT-Large	77.3 (2.2)	88.7 (4.3)	96.3 (0.9)	98.5 (0.7)	76.4 (2.0)	90.2 (2.9)	96.3 (1.0)	98.8 (0.7)
spring	Static emb.	Fasttext-B	33.0 (6.8)	43.8 (8.2)	46.4 (7.4)	67.0 (2.2)	35.4 (0.5)	32.8 (0.7)	35.8 (3.6)	66.7 (3.1)
		Fasttext-C	46.0 (14.6)	57.6 (4.0)	73.5 (3.7)	83.7 (2.8)	45.0 (9.0)	64.2 (3.3)	76.5 (3.5)	86.1 (2.8)

Table 16
(*continued*)

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
	1NN	BERT-Base	94.4 (2.0)	97.2 (0.5)	97.1 (0.3)	97.3 (0.1)	94.6 (2.2)	97.3 (0.9)	97.0 (0.4)	97.3 (0.1)
		BERT-Large	97.1 (1.5)	97.9 (0.5)	97.6 (0.4)	97.7 (0.2)	96.8 (1.6)	97.8 (0.2)	97.5 (0.3)	97.8 (0.1)
	Fine-Tuning	BERT-Base	75.2 (4.7)	92.9 (0.3)	96.0 (0.5)	95.3 (0.5)	73.9 (4.3)	92.9 (0.2)	96.1 (0.5)	95.2 (0.6)
		BERT-Large	80.3 (10.1)	94.2 (2.7)	97.0 (0.7)	97.2 (0.2)	77.1 (12.6)	94.4 (2.4)	97.1 (0.6)	97.1 (0.4)
chair	Static emb.	Fasttext-B	62.8 (9.0)	73.8 (6.6)	74.4 (4.5)	74.4 (5.2)	58.4 (6.9)	68.4 (5.1)	68.4 (4.3)	72.1 (4.2)
		Fasttext-C	76.2 (8.0)	75.4 (4.5)	81.3 (2.0)	83.6 (2.4)	64.1 (12.8)	72.9 (0.2)	75.8 (0.9)	81.7 (2.2)
	1NN	BERT-Base	88.7 (7.1)	95.9 (0.7)	95.9 (0.4)	95.6 (0.4)	84.2 (11.6)	94.5 (0.5)	94.5 (0.3)	94.3 (0.3)
		BERT-Large	82.3 (19.0)	94.6 (1.3)	96.2 (0.0)	95.9 (0.4)	84.4 (14.1)	93.5 (0.9)	94.7 (0.0)	94.5 (0.3)
	Fine-Tuning	BERT-Base	84.1 (11.3)	91.0 (5.7)	95.6 (0.7)	96.7 (0.4)	75.6 (21.8)	90.1 (5.8)	94.9 (0.8)	96.1 (0.4)
		BERT-Large	72.1 (11.9)	93.3 (1.5)	94.4 (2.0)	95.1 (1.5)	65.4 (12.9)	92.4 (1.7)	93.6 (2.1)	94.4 (1.6)
hood	Static emb.	Fasttext-B	56.1 (11.3)	33.3 (16.2)	47.6 (18.7)	–	48.8 (4.1)	42.1 (6.3)	51.4 (5.1)	–
		Fasttext-C	66.3 (5.0)	77.6 (2.1)	86.6 (5.0)	–	61.5 (6.2)	73.7 (4.7)	82.1 (6.8)	–
	1NN	BERT-Base	96.8 (3.0)	98.4 (0.6)	98.8 (0.0)	–	94.8 (5.9)	98.0 (0.7)	98.5 (0.0)	–
		BERT-Large	96.3 (4.3)	99.2 (0.6)	99.6 (0.6)	–	92.7 (9.3)	99.0 (0.7)	99.5 (0.7)	–
	Fine-Tuning	BERT-Base	86.6 (3.6)	95.5 (2.5)	97.6 (1.7)	–	79.0 (6.4)	94.4 (3.0)	96.7 (2.4)	–
		BERT-Large	89.8 (5.8)	96.3 (1.0)	98.8 (1.0)	–	80.1 (16.2)	95.1 (1.5)	98.0 (1.7)	–
seal	Static emb.	Fasttext-B	29.9 (1.0)	31.6 (3.2)	39.9 (10.0)	60.2 (4.2)	25.0 (0.0)	25.7 (1.0)	39.8 (5.6)	57.4 (1.1)
		Fasttext-C	46.4 (8.1)	64.6 (4.0)	73.7 (2.3)	82.1 (1.6)	43.6 (11.2)	67.7 (5.1)	79.0 (2.4)	85.4 (2.8)
	1NN	BERT-Base	91.5 (6.8)	96.1 (0.7)	96.4 (0.4)	96.6 (0.3)	89.4 (11.0)	97.0 (0.6)	97.3 (0.3)	97.5 (0.3)
		BERT-Large	96.1 (1.8)	97.0 (0.7)	98.0 (0.6)	98.2 (0.1)	96.5 (1.5)	97.7 (0.6)	98.4 (0.5)	98.6 (0.1)
	Fine-Tuning	BERT-Base	79.6 (7.9)	95.0 (1.0)	94.4 (1.5)	96.6 (0.5)	72.3 (12.5)	90.0 (1.5)	88.7 (3.4)	92.3 (0.8)
		BERT-Large	76.2 (12.4)	94.0 (0.9)	94.6 (2.4)	97.1 (0.6)	68.0 (16.4)	89.0 (3.5)	88.3 (4.4)	94.0 (1.4)
bow	Static emb.	Fasttext-B	29.8 (16.7)	41.2 (20.6)	40.2 (9.1)	63.3 (4.3)	39.1 (7.6)	35.2 (2.3)	39.5 (7.9)	64.8 (1.5)
		Fasttext-C	52.3 (8.1)	62.6 (4.6)	79.4 (0.8)	87.6 (1.2)	49.3 (7.2)	60.5 (6.1)	76.4 (1.8)	87.1 (1.5)
	1NN	BERT-Base	86.5 (1.3)	91.8 (2.7)	95.5 (0.2)	95.3 (0.4)	80.7 (4.2)	88.6 (2.0)	93.7 (1.5)	95.0 (0.5)
		BERT-Large	87.4 (1.7)	94.9 (2.3)	97.4 (0.6)	96.4 (1.0)	84.3 (4.2)	93.3 (4.5)	97.8 (0.5)	96.7 (1.0)

Table 16
(*continued*)

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
club	Fine-Tuning	BERT-Base	83.1 (3.1)	89.5 (4.1)	94.0 (0.8)	96.1 (0.2)	73.2 (6.4)	85.6 (4.9)	93.1 (1.5)	95.3 (0.4)
		BERT-Large	78.8 (13.6)	91.0 (3.9)	96.7 (0.7)	97.5 (0.6)	73.1 (12.0)	91.0 (2.2)	96.9 (0.9)	97.5 (1.1)
	Static emb.	Fasttext-B	35.0 (11.8)	26.2 (19.3)	23.8 (5.5)	56.1 (4.1)	31.5 (1.0)	32.6 (1.0)	37.1 (1.5)	54.4 (1.2)
		Fasttext-C	35.2 (8.1)	47.7 (5.4)	60.2 (3.8)	74.6 (2.8)	37.4 (1.9)	52.3 (3.6)	61.7 (2.5)	79.0 (2.3)
	1NN	BERT-Base	58.3 (5.5)	80.2 (1.9)	81.2 (1.1)	81.2 (0.8)	70.5 (2.8)	84.9 (2.3)	84.9 (1.7)	84.7 (1.0)
		BERT-Large	54.1 (10.5)	82.8 (3.5)	83.8 (1.8)	84.5 (0.2)	69.0 (7.7)	87.4 (3.2)	88.9 (1.7)	88.5 (0.3)
	Fine-Tuning	BERT-Base	52.5 (6.1)	68.5 (8.9)	80.5 (1.5)	84.2 (0.8)	50.0 (6.2)	68.3 (10.0)	79.9 (1.9)	83.4 (0.8)
		BERT-Large	51.3 (7.3)	71.8 (1.9)	81.2 (3.9)	83.7 (1.9)	49.8 (6.5)	69.5 (2.7)	80.8 (4.1)	83.4 (1.5)
trunk	Static emb.	Fasttext-B	32.9 (16.4)	21.2 (0.6)	45.9 (17.7)	66.7 (3.4)	34.0 (2.1)	35.4 (2.0)	43.5 (7.4)	67.2 (2.8)
		Fasttext-C	65.4 (6.8)	63.2 (7.4)	76.6 (2.1)	82.7 (3.7)	65.3 (8.8)	67.0 (7.9)	78.3 (0.7)	87.4 (2.8)
	1NN	BERT-Base	77.9 (18.4)	84.8 (7.1)	94.8 (1.8)	95.2 (2.2)	84.1 (12.1)	91.2 (4.6)	97.2 (1.0)	97.4 (1.2)
		BERT-Large	83.1 (20.2)	96.1 (1.1)	96.5 (1.2)	98.3 (0.6)	89.7 (11.5)	97.9 (0.6)	98.1 (0.7)	99.1 (0.3)
	Fine-Tuning	BERT-Base	72.7 (16.7)	89.2 (5.4)	93.9 (1.6)	97.0 (1.6)	72.5 (11.2)	89.1 (4.6)	93.5 (1.6)	96.7 (1.8)
		BERT-Large	79.2 (14.7)	91.3 (2.4)	95.2 (1.2)	96.1 (1.8)	79.9 (11.7)	90.9 (2.4)	94.9 (1.4)	95.7 (2.0)
	Static emb.	Fasttext-B	20.5 (7.2)	8.9 (3.6)	38.5 (10.3)	—	25.9 (0.8)	25.0 (0.0)	38.8 (1.9)	—
		Fasttext-C	40.1 (19.0)	60.4 (10.8)	71.8 (3.2)	—	39.4 (7.1)	61.5 (5.5)	74.6 (2.9)	—
square	1NN	BERT-Base	70.2 (7.7)	83.3 (7.8)	90.7 (3.0)	—	74.1 (10.3)	83.7 (5.1)	88.6 (2.2)	—
		BERT-Large	74.9 (13.0)	84.7 (8.3)	93.7 (1.2)	—	79.4 (4.6)	84.4 (5.6)	89.0 (4.0)	—
	Fine-Tuning	BERT-Base	64.9 (11.3)	75.4 (11.3)	85.2 (3.6)	—	56.7 (4.3)	71.0 (8.1)	81.3 (4.0)	—
		BERT-Large	63.9 (17.0)	81.0 (10.2)	87.3 (2.0)	—	57.7 (7.3)	76.9 (8.2)	82.9 (1.8)	—
arm	Static emb.	Fasttext-B	53.7 (11.1)	60.0 (3.9)	53.3 (8.9)	80.3 (2.0)	58.6 (1.9)	61.4 (3.5)	62.3 (3.3)	79.9 (2.5)
		Fasttext-C	79.1 (7.1)	85.4 (7.0)	90.9 (4.7)	95.7 (0.5)	85.1 (5.1)	86.6 (5.5)	91.6 (3.0)	95.1 (0.8)
	1NN	BERT-Base	99.4 (0.00)	99.4 (0.0)	99.4 (0.0)	99.4 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)
		BERT-Large	99.4 (0.00)	99.4 (0.0)	99.4 (0.0)	99.4 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)
	Fine-Tuning	BERT-Base	96.3 (2.8)	99.4 (0.0)	99.4 (0.0)	99.4 (0.0)	95.0 (3.9)	99.2 (0.0)	99.2 (0.0)	99.2 (0.0)
		BERT-Large	97.4 (2.1)	98.8 (0.9)	99.4 (0.0)	99.4 (0.0)	96.4 (2.9)	98.4 (1.1)	99.2 (0.0)	99.2 (0.0)

Table 16
(*continued*)

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
digit	Static emb.	Fasttext-B	45.2 (5.1)	54.0 (16.8)	43.7 (9.0)	—	49.0 (5.1)	69.4 (11.1)	62.8 (5.0)	—
		Fasttext-C	69.8 (9.2)	84.9 (8.1)	87.3 (5.9)	—	63.3 (3.8)	85.0 (5.5)	89.2 (2.7)	—
	1NN	BERT-Base	96.8 (2.2)	99.2 (1.1)	100 (0.0)	—	92.6 (5.2)	98.1 (2.6)	100 (0.0)	—
		BERT-Large	94.4 (6.3)	100 (0.0)	100 (0.0)	—	87.0 (14.6)	100 (0.0)	100 (0.0)	-
	Fine-Tuning	BERT-Base	96.0 (2.2)	100 (0.0)	100 (0.0)	—	93.6 (4.1)	100 (0.0)	100 (0.0)	—
		BERT-Large	95.2 (5.1)	98.4 (1.1)	100 (0.0)	—	91.2 (10.0)	97.5 (1.7)	100 (0.0)	—
bass	Static emb.	Fasttext-B	27.8 (8.9)	22.2 (0.5)	33.5 (15.3)	60.7 (3.6)	39.2 (3.6)	36.8 (2.6)	39.2 (6.7)	71.4 (1.1)
		Fasttext-C	37.1 (8.2)	49.8 (5.3)	65.1 (6.2)	78.9 (3.2)	49.4 (3.9)	57.5 (3.3)	67.1 (1.8)	78.3 (2.2)
	1NN	BERT-Base	65.6 (17.9)	75.2 (7.1)	70.4 (5.7)	77.2 (2.1)	60.4 (3.2)	71.8 (5.8)	71.2 (1.6)	77.2 (0.5)
		BERT-Large	70.2 (17.8)	76.9 (6.5)	77.1 (4.5)	83.4 (4.2)	70.3 (4.6)	78.6 (4.5)	80.3 (1.9)	83.4 (0.6)
	Fine-Tuning	BERT-Base	63.0 (12.3)	67.8 (5.5)	82.5 (1.8)	86.5 (1.2)	54.2 (2.6)	62.2 (3.9)	71.1 (1.4)	76.2 (0.8)
		BERT-Large	79.4 (15.5)	70.4 (10.0)	76.4 (7.5)	88.1 (2.8)	67.6 (7.7)	65.0 (5.3)	68.7 (5.2)	78.5 (2.7)
yard	Static emb.	Fasttext-B	48.6 (10.8)	35.6 (2.6)	40.3 (12.3)	—	56.0 (7.0)	54.6 (7.5)	61.0 (5.6)	—
		Fasttext-C	63.4 (26.2)	74.1 (9.8)	83.8 (9.2)	—	62.3 (13.9)	78.5 (7.2)	86.7 (6.2)	—
	1NN	BERT-Base	52.8 (15.8)	70.8 (11.9)	77.3 (2.9)	—	68.4 (13.0)	82.8 (7.1)	86.6 (1.7)	—
		BERT-Large	65.3 (20.5)	81.0 (14.3)	85.6 (7.6)	—	79.5 (12.1)	88.8 (8.4)	91.5 (4.5)	—
	Fine-Tuning	BERT-Base	56.0 (13.6)	67.1 (15.9)	92.1 (4.6)	—	48.3 (10.5)	62.1 (13.1)	87.8 (6.2)	—
		BERT-Large	48.6 (8.6)	82.9 (9.6)	90.7 (5.7)	—	46.7 (7.2)	77.4 (9.7)	85.9 (7.8)	—
pound	Static emb.	Fasttext-B	57.7 (20.7)	39.9 (18.3)	40.5 (3.4)	—	57.3 (7.0)	51.7 (2.6)	55.1 (2.0)	—
		Fasttext-C	61.2 (20.5)	58.8 (5.8)	68.7 (5.1)	—	57.7 (7.5)	62.3 (2.9)	73.7 (3.8)	—
	1NN	BERT-Base	61.9 (19.0)	66.3 (14.5)	77.7 (9.1)	—	55.1 (6.6)	81.2 (8.1)	86.1 (4.1)	—
		BERT-Large	69.4 (26.0)	69.4 (8.5)	82.1 (5.1)	—	59.4 (3.0)	81.5 (6.1)	90.0 (2.8)	—
	Fine-Tuning	BERT-Base	61.9 (10.2)	63.6 (16.6)	74.2 (9.7)	—	45.1 (4.7)	52.8 (10.7)	64.5 (8.2)	—
		BERT-Large	64.6 (17.2)	59.1 (2.6)	81.1 (9.4)	—	47.9 (5.9)	51.4 (1.5)	70.0 (8.0)	—

Table 16
(*continued*)

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
deck	Static emb.	Fasttext-B	54.6 (17.2)	73.1 (10.8)	71.0 (10.5)	–	51.4 (7.9)	54.7 (4.3)	62.4 (2.6)	–
		Fasttext-C	68.4 (32.0)	66.0 (13.0)	77.8 (3.3)	–	61.0 (2.1)	61.9 (4.1)	72.6 (10.0)	–
	1NN	BERT-Base	81.8 (12.0)	85.2 (8.1)	86.9 (1.4)	–	81.4 (9.0)	81.0 (4.6)	90.7 (2.8)	–
		BERT-Large	76.4 (15.9)	87.5 (2.5)	90.9 (0.8)	–	78.5 (2.9)	84.5 (5.3)	95.1 (0.4)	–
	Fine-Tuning	BERT-Base	86.9 (7.3)	87.9 (1.4)	86.9 (2.2)	–	70.8 (11.9)	69.9 (1.3)	70.3 (3.6)	–
		BERT-Large	77.4 (17.8)	88.6 (2.4)	90.6 (2.7)	–	55.2 (10.4)	63.8 (12.2)	77.7 (4.0)	–
bank	Static emb.	Fasttext-B	46.5 (12.1)	46.9 (4.6)	51.0 (8.8)	77.8 (5.5)	56.8 (2.0)	64.9 (1.2)	62.7 (2.7)	72.5 (3.7)
		Fasttext-C	38.4 (6.8)	70.1 (11.9)	80.7 (2.8)	88.2 (4.6)	61.9 (0.3)	72.8 (4.6)	79.1 (3.4)	85.9 (2.4)
	1NN	BERT-Base	98.8 (0.7)	99.4 (0.5)	99.7 (0.1)	99.8 (0.0)	94.3 (3.3)	95.4 (4.9)	97.7 (0.1)	97.7 (0.0)
		BERT-Large	99.0 (0.7)	99.5 (0.1)	99.9 (0.1)	99.9 (0.1)	95.9 (3.5)	97.6 (1.8)	99.2 (1.1)	99.2 (1.1)
	Fine-Tuning	BERT-Base	91.9 (5.7)	97.8 (1.4)	97.9 (0.8)	99.0 (0.1)	76.1 (9.4)	89.4 (5.2)	90.5 (3.3)	95.3 (0.5)
		BERT-Large	58.0 (28.9)	98.6 (0.7)	99.5 (0.1)	98.6 (0.6)	52.3 (28.3)	92.3 (4.0)	97.3 (0.5)	93.5 (2.4)
pitcher	Static emb.	Fasttext-B	92.1 (2.2)	82.7 (9.3)	82.8 (1.3)	–	69.2 (8.0)	73.4 (10.1)	82.4 (4.3)	–
		Fasttext-C	96.5 (4.2)	95.9 (1.8)	91.2 (3.0)	–	84.2 (13.5)	94.1 (0.9)	94.3 (3.0)	–
	1NN	BERT-Base	100 (0.0)	99.9 (0.1)	99.8 (0.0)	–	100 (0.0)	99.9 (0.0)	99.9 (0.0)	–
		BERT-Large	100 (0.0)	100 (0.0)	99.9 (0.0)	–	100 (0.0)	100 (0.0)	100 (0.0)	–
	Fine-Tuning	BERT-Base	98.6 (0.5)	98.9 (0.6)	97.2 (1.1)	–	70.6 (5.3)	76.2 (10.2)	62.8 (4.0)	–
		BERT-Large	97.1 (2.3)	98.7 (1.2)	98.5 (1.1)	–	70.5 (17.3)	77.3 (13.8)	74.8 (13.6)	–

Table 17
Sense definitions in the CoarseWSD-20 data set. Each sense is accompanied with an example usage from the data set. Sense IDs correspond to the current Wikipedia page of each sense by the date of the submission.

Word	Sense #	Sense ID	Definition (1st sentence from Wikipedia)	Example usage (tokenized)
Crane	$crane_1$	crane (machine)	A crane is a type of machine, generally equipped with a hoist rope, wire ropes or chains, and sheaves, that can be used both to lift and lower materials and to move them horizontally.	launching and recovery is accomplished with the assistance of a shipboard crane .
	$crane_2$	crane (bird)	Cranes are a family, the Gruidae, of large, long-legged, and long-necked birds in the group Gruiformes.	tibet hosts species of wolf , wild donkey , crane, vulture , hawk , geese , snake , and buffalo .
Java	$java_1$	java	Java is an island of Indonesia, bordered by the Indian Ocean on the south and the Java Sea on the north.	in indonesia , only sumatra , borneo, and papua are larger in territory , and only java and sumatra have larger populations .
	$java_2$	java (programming language)	Java is a general-purpose programming language that is class- based, object-oriented, and designed to have as few implementation dependencies as possible.	examples include the programming languages perl , java and lua .
Apple	$apple_1$	apple inc.	Apple Inc. is an American multinational technology company headquartered in Cupertino, California, that designs, develops, and sells consumer electronics, computer software, and online services.	shopify released a free mobile app on the apple app store on may 13, 2010.
	$apple_2$	apple	An apple is an edible fruit produced by an apple tree.	cherry, apple, pear, peach and apricot trees are available.
Mole	$mole_1$	mole (animal)	Moles are small mammals adapted to a subterranean lifestyle (i.e., fossorial).	its primary prey consists of mice, rat, squirrel, chipmunk, shrew, mole and rabbits.
	$mole_2$	mole (espionage)	In espionage jargon, a mole is a long-term spy who is recruited before having access to secret intelligence, subsequently managing to get into the target organization.	philip meets claudia where she tells him that there is a mole working for the fbi.
	$mole_3$	mole (unit)	The mole (symbol: mol) is the unit of measurement for amount of substance in the International System of Units (SI).	so the specific heat of a classical solid is always 3k per atom , or in chemistry units, 3r per mole of atoms .
	$mole_4$	mole sauce	Mole is a traditional marinade and sauce originally used in Mexican cuisine.	food such as cake, chicken with mole, hot chocolate, coffee, and atole are served .

Table 17
(continued)

Word	Sense #	Sense ID	Definition (1st sentence from Wikipedia)	Example usage (tokenized)
Spring	$mole_5$	mole (architecture)	A mole is a massive structure, usually of stone, used as a pier, breakwater, or a causeway between places separated by water.	the islands of pomgues and ratonneau are connected by a mole built in 1822.
	$spring_1$	spring (hydrology)	A spring is a point at which water flows from an aquifer to the Earth's surface. It is a component of the hydrosphere.	the village was famous for its mineral water spring used for healing in sanatorium , including the hawthorne and lithia springs .
	$spring_2$	spring (season)	Spring, also known as springtime, is one of the four temperate seasons, succeeding winter and preceding summer.	the species is most active during the spring and early summer although it may be seen into late june .
	$spring_3$	spring (device)	A spring is an elastic object that stores mechanical energy.	often spring are used to reduce backlash of the mechanism .
Chair	$chair_1$	chairman	The chairperson (also chair, chairman, or chairwoman) is the presiding officer of an organized group such as a board, committee, or deliberative assembly.	gan is current chair of the department of environmental sciences at university of california , riverside .
	$chair_2$	chair	One of the basic pieces of furniture, a chair is a type of seat.	a typical western living room may contain furnishings such as a sofa , chair , occasional table , and bookshelves , electric lamp , rugs , or other furniture .
Hood	$hood_1$	hood (comics)	Hood (real name Parker Robbins) is a fictional character, a supervillain, and a crime boss appearing in American comic books published by Marvel Comics.	the hood has hired him as part of his criminal organization to take advantage of the split in the superhero community caused by the super-human registration act .
	$hood_2$	hood (vehicle)	The hood (North American English) or bonnet (Commonwealth English excluding Canada) is the hinged cover over the engine of motor vehicles that allows access to the engine compartment, or trunk (boot in Commonwealth English) on rear-engine and some mid-engine vehicles) for maintenance and repair.	european versions of the car also had an air intake on the hood .
	$hood_3$	hood (headgear)	A hood is a kind of headgear that covers most of the head and neck, and sometimes the face.	in some sauna suits , the jacket also includes a hood to provide additional retention of body heat.

Table 17
(continued)

Word	Sense #	Sense ID	**Definition** (1st sentence from Wikipedia)	**Example usage** (tokenized)
Seal	$seal_1$	pinniped	Pinnipeds, commonly known as seals, are a widely distributed and diverse clade of carnivorous, fin-footed, semiaquatic marine mammals.	animals such as shark , stingray , weever fish , seal and jellyfish can sometimes present a danger .
	$seal_2$	seal (musician)	Henry Olusegun Adeola Samue (born 19 February 1963), known professionally as Seal, is a British singer-songwriter.	she was married to english singer seal from 2005 until 2012 .
	$seal_3$	seal (emblem)	A seal is a device for making an impression in wax, clay, paper, or some other medium, including an embossment on paper, and is also the impression thus made.	each level must review , add information as necessary , and stamp or seal that the submittal was examined and approved by that party .
	$seal_4$	seal (mechanical)	A mechanical seal is a device that helps join systems or mechanisms together by preventing leakage (e.g. in a pumping system), containing pressure, or excluding contamination.	generally speaking , standard ball joints will outlive sealed ones because eventually the seal will break , causing the joint to dry out and rust .
Bow	bow_1	bow (ship)	The bow is the forward part of the hull of a ship or boat.	the stem is the most forward part of a boat or ship's bow and is an extension of the keel itself .
	bow_2	bow and arrow	The bow and arrow is a ranged weapon system consisting of an elastic launching device (bow) and long-shafted projectiles (arrows).	bow and arrow used in warfare .
	bow_3	bow (music)	In music, a bow is a tensioned stick which has hair (usually horse-tail hair) coated in rosin (to facilitate friction) affixed to it.	horsehair is used for brush , the bow of musical instruments and many other things .
Club	$club_1$	club	A club is an association of people united by a common interest or goal.	this is a partial list of women's association football club teams from all over the world sorted by confederation .
	$club_2$	nightclub	A nightclub, music club, or club, is an entertainment venue and bar that usually operates late into the night.	although several of his tracks were club hits, he had limited chart success.
	$club_3$	club (weapon)	A club (also known as a cudgel, baton, bludgeon, truncheon, cosh, nightstick or impact weapon) is among the simplest of all weapons: a short staff or stick, usually made of wood, wielded as a weapon since prehistoric times.	before their adoption of guns, the plains indians hunted with spear, bows and arrows, and various forms of club.

Table 17
(continued)

Word	Sense #	Sense ID	**Definition** (1st sentence from Wikipedia)	**Example usage** (tokenized)
Trunk	$trunk_1$	trunk (botany)	In botany, the trunk (or bole) is the stem and main wooden axis of a tree.	its leaves are different from the leaves of true palms, and unlike true palms it does not develop a woody trunk.
	$trunk_2$	trunk (automobile)	The trunk (North American English), boot (British English), dickey (Indian English) (also spelled dicky or diggy) or compartment (South-East Asia) of a car is the vehicle's main storage or cargo compartment.	unlike the bmw x5, the x-coupe had an aluminium body, a trunk opening downwards and two doors that swing outward.
	$trunk_3$	trunk (anatomy)	The torso or trunk is an anatomical term for the central part orx core of many animal bodies (including humans) from which extend the neck and limbs.	surface projections of the major organs of the trunk, using the vertebral column and rib cage as main reference points of superficial anatomy.
Square	$square_1$	square	In geometry, a square is a regular quadrilateral, which means that it has four equal sides and four equal angles (90-degree angles, or 100-gradian angles or right angles).	similarly , a square with all sides of length has the perimeter and the same area as the rectangle.
	$square_2$	square (company)	Square Co., Ltd. was a Japanese video game company founded in September 1986 by Masafumi Miyamoto. It merged with Enix in 2003 to form Square Enix.	video game by square , features the orbital elevator " a.t.l.a.s. " .
	$square_3$	town square	A town square is an open public space commonly found in the heart of a traditional town used for community gatherings.	here is a partial list of notable expressways, tunnel, bridge, road, avenues, street, crescent, square and bazaar in hong kong.
	$square_4$	square number	In mathematics, a square number or perfect square is an integer that is the square of an integer.	in mathematics eighty-one is the square of 9 and the fourth power of 3.
Arm	arm_1	arm architecture	Arm (previously officially written all caps as ARM and usually written as such today), previously Advanced RISC Machine, originally Acorn RISC Machine, is a family of reduced instruction set computing (RISC) architectures for computer processors, configured for various environments.	windows embedded compact is available for arm, mips, superh and x86 processor architectures.

Table 17
(continued)

Word	Sense #	Sense ID	Definition (1st sentence from Wikipedia)	Example usage (tokenized)
Digit	arm_2	arm	In human anatomy, the arm is the part of the upper limb between the glenohumeral joint (shoulder joint) and the elbow joint.	on the human body, the limb can be divided into segments, such as the arm and the forearm of the upper limb, and the thigh and the leg of the lower slimb.
	$digit_1$	numerical digit	A numerical digit is a single symbol (such as "2" or "5") used alone, or in combinations (such as "25"), to represent numbers (such as the number 25) according to some positional numeral systems.	it uses the digit 0 , 1 , 2 and 3 to represent any real number.
	$digit_2$	digit (anatomy)	A digit is one of several most distal parts of a limb, such as fingers or toes, present in many vertebrates.	a finger is a limb of the human body and a type of digit, an organ of and found in the hand of human and other primate.
Bass	$bass_1$	bass (guitar)	The bass guitar, electric bass, or simply bass, is the lowest-pitched member of the guitar family.	the band decided to continue making music after thirsk's death, and brought in bass guitarist randy bradbury from one hit wonder.
	$bass_2$	bass (voice type)	A bass is a type of classical male singing voice and has the lowest vocal range of all voice types.	he is known for his distinctive and untrained bass voice.
	$bass_3$	double bass	The double bass, also known simply as the bass (or by other names), is the largest and lowest-pitched bowed (or plucked) string instrument in the modern symphony orchestra.	his instruments were the bass and the tuba.
Yard	$yard_1$	yard	The yard (abbreviation: yd) is an English unit of length, in both the British imperial and US customary systems of measurement, that comprises 3 feet or 36 inches.	accuracy is sufficient for hunting small game at ranges to 50 yard.
	$yard_2$	yard (sailing)	A yard is a spar on a mast from which sails are set.	aubrey improves sophie sailing qualities by adding a longer yard which allows him to spread a larger mainsail.

Table 17
(continued)

Word	Sense #	Sense ID	Definition (1st sentence from Wikipedia)	Example usage (tokenized)
Pound	$pound_1$	pound (mass)	The pound or pound-mass is a unit of mass used in the imperial, United States customary and other systems of measurement.	it is approximately 16.38 kilogram (36.11 pound).
	$pound_2$	pound (currency)	A pound is any of various units of currency in some nations.	in english, the maltese currency was referred to as the pound originally and for many locals this usage continued.
Deck	$deck_1$	deck (ship)	A deck is a permanent covering over a compartment or a hull of a ship.	the protective deck was thick and ran the full length of the ship.
	$deck_2$	deck (building)	In architecture, a deck is a flat surface capable of supporting weight, similar to a floor, but typically constructed outdoors, often elevated from the ground, and usually connected to a building.	typically , it is a wooden deck near a hiking trail that provides the hikers a clean and even place to sleep.
Bank	$bank_1$	bank	A bank is a financial institution that accepts deposits from the public and creates a demand deposit, while simultaneously making loans.	the bank , which loans money to the player after they have a house for collateral .
	$bank_2$	bank (geography)	In geography, a bank is the land alongside a body of water.	singapore's first market was located at the south bank of the singapore river.
Pitcher	$pitcher_1$	pitcher	In baseball, the pitcher is the player who throws the baseball from the pitcher's mound toward the catcher to begin each play, with the goal of retiring a batter, who attempts to either make contact with the pitched ball or draw a walk.	kasey garret olemberger (born march 18 , 1978) is an italian american professional baseball pitcher.
	$pitcher_2$	pitcher (container)	In American English, a pitcher is a container with a spout used for storing and pouring liquids.	pottery was found as grave goods, including combinations of pitcher and cup.

Acknowledgments

We would like to thank Claudio Delli Bovi and Miguel Ballesteros for early pre-BERT discussions on the topic of ambiguity and language models. We would also like to thank the anonymous reviewers for their comments and suggestions that helped improve the article. Daniel Loureiro is supported by the European Union and Fundação para a Ciência e Tecnologia through contract DFA/BD/9028/2020 (Programa Operacional Regional Norte). Jose Camacho-Collados is supported by a UKRI Future Leaders Fellowship.

References

Agirre, Eneko, Oier López de Lacalle, and Aitor Soroa. 2018. The risk of sub-optimal use of open source NLP software: UKB is inadvertently state-of-the-art in knowledge-based WSD. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 29–33, Melbourne. https://doi.org/10.18653/v1/W18-2505

Agirre, Eneko, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84. https://doi.org/10.1162/COLI_a_00164

Agirre, Eneko, Lluís Màrquez, and Richard Wicentowski, editors. 2007. Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). Prague. https://doi.org/10.3115/1621474.1621476

Aina, Laura, Kristina Gulordava, and Gemma Boleda. 2019. Putting words in context: LSTM language models and lexical ambiguity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3348, Florence. https://doi.org/10.18653/v1/P19-1324

Akbik, Alan, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, NM.

Amrami, Asaf and Yoav Goldberg. 2018. Word sense induction with neural biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4867, Brussels. https://doi.org/10.18653/v1/D18-1523

Apidianaki, Marianna. 2008. Translation-oriented word sense induction based on parallel corpora. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), pages 3269–3275, Marrakech.

Banerjee, Satanjeev and Ted Pedersen. 2003. Extended gloss overlap as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco.

Bansal, Mohit, John DeNero, and Dekang Lin. 2012. Unsupervised translation sense clustering. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 773–782, Montréal.

Basile, Pierpaolo, Annalina Caputo, and Giovanni Semeraro. 2014. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin.

Belinkov, Yonatan and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72. https://doi.org/10.1162/tacl_a_00254

Bennett, Andrew, Timothy Baldwin, Jey Han Lau, Diana McCarthy, and Francis Bond. 2016. LexSemTm: A semantic data set based on all-words unsupervised sense distribution learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1524, Berlin. https://doi.org/10.18653/v1/P16-1143

Bevilacqua, Michele and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. https://doi.org/10.18653/v1/2020.acl-main.255

Blevins, Terra and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017,

Online. https://doi.org/10.18653/v1/2020.acl-main.95

Bond, Francis and Ryan Foster. 2013. Linking and extending an open multilingual WordNet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Camacho-Collados, Jose and Roberto Navigli. 2017. BabelDomains: Large-scale domain labeling of lexical resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 223–228, Valencia. https://doi.org/10.18653/v1/E17-2036

Camacho-Collados, Jose and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788. https://doi.org/10.1613/jair.1.11259

Chaplot, Devendra Singh and Ruslan Salakhutdinov. 2018. Knowledge-based word sense disambiguation using topic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 5062–5069, New Orleans, LA.

Chronis, Gabriella and Katrin Erk. 2020. When is a bishop not like a rook? When it's like a rabbi! Multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. https://doi.org/10.18653/v1/2020.conll-1.17

Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence. https://doi.org/10.18653/v1/W19-4828

Conneau, Alexis, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne. https://doi.org/10.18653/v1/P18-1198

de Vries, Wietse, Andreas van Cranenburgh, and Malvina Nissim. 2020. What's so special about BERT's layers? A closer look at the NLP pipeline in monolingual and multilingual models. *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4339–4350, Online.

Delli Bovi, Claudio, Jose Camacho-Collados, Alessandro Raganato, and Roberto Navigli. 2017. EuroSense: Automatic harvesting of multilingual sense annotations from parallel text. In *Proceedings of ACL*, volume 2, pages 594–600, Vancouver. https://doi.org/10.18653/v1/P17-2094

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN.

Edmonds, Philip and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse.

Ethayarajh, Kawin. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong. https://doi.org/10.18653/v1/D19-1006

Ettinger, Allyson. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48. https://doi.org/10.1162/tacl_a_00298

Federmeier, Kara D., Jessica B. Segal, Tania Lombrozo, and Marta Kutas. 2000. Brain

responses to nouns, verbs and class-ambiguous words in context. *Brain*, 123(12):2552–2566. https://doi.org/10.1093/brain/123.12.2552, PubMed: 11099456

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Database*, MIT Press, Cambridge, MA. https://doi.org/10.7551/mitpress/7287.001.0001

Flekova, Lucie and Iryna Gurevych. 2016. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2029–2041, Berlin. https://doi.org/10.18653/v1/P16-1191

Gale, William A., Kenneth Church, and David Yarowsky. 1992a. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 249–256, Newark, DE. https://doi.org/10.3115/981967.981999

Gale, William A., Kenneth W. Church, and David Yarowsky. 1992b. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5):415–439. https://doi.org/10.1007/BF00136984

Goldberg, Yoav. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Haber, Janosch and Massimo Poesio. 2020. Word sense distance in human similarity judgements and contextualised word embeddings. In *Proceedings of the Probability and Meaning Conference (PaM 2020)*, pages 128–145, Barcelona.

Hardt, Moritz, Eric Price, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3315–3323, Curran Associates, Inc.

Hauer, Bradley and Grzegorz Kondrak. 2020. One homonym per translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7895–7902, New York, NY. https://doi.org/10.1609/aaai.v34i05.6296

Hewitt, John and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, MN.

Hovy, Eduard H., Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194:2–27. https://doi.org/10.1016/j.artint.2012.10.002

Huang, Luyao, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3507–3512, Hong Kong. https://doi.org/10.18653/v1/D19-1355

Iacobacci, Ignacio, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin. https://doi.org/10.18653/v1/P16-1085

Ide, Nancy, Collin Baker, Christiane Fellbaum, Charles Fillmore, and Rebecca Passonneau. 2008. MASC: The manually annotated sub-corpus of American English. *6th International Conference on Language Resources and Evaluation, LREC 2008*, pages 2455–2460, Miyazaki.

Ilievski, Filip, Piek Vossen, and Stefan Schlobach. 2018. Systematic study of long tail phenomena in entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 664–674, Santa Fe, NM.

Jawahar, Ganesh, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence. https://doi.org/10.18653/v1/P19-1356

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia. https://doi.org/10.18653/v1/E17-2068

Jurgens, David and Mohammad Taher Pilehvar. 2015. Reserating the awesometastic: An automatic extension of the WordNet taxonomy for novel terms. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1459–1465, Denver, CO. https://doi.org/10.3115/v1/N15-1169

Kumar, Sawan, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence. https://doi.org/10.18653/v1/P19-1568

Kuncoro, Adhiguna, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne. https://doi.org/10.18653/v1/P18-1132

Lacerra, Caterina, Michele Bevilacqua, Tommaso Pasini, and Roberto Navigli. 2020. CSI: A coarse sense inventory for 85% word sense disambiguation. In *Proceedings of the 34th Conference on Artificial Intelligence*, pages 8123–8130, New York, NY. https://doi.org/10.1609/aaai.v34i05.6324

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, OpenReview.net.

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual Conference on Systems Documentation*, pages 24–26, Toronto. https://doi.org/10.1145/318723.318728

Levine, Yoav, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. https://doi.org/10.18653/v1/2020.acl-main.423

Ling, Xiao, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328. https://doi.org/10.1162/tacl_a_00141

Linzen, Tal, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535. https://doi.org/10.1162/tacl_a_00115

Liu, Nelson F., Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, MN.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loureiro, Daniel and Jose Camacho-Collados. 2020. Don't neglect the obvious: On the role of unambiguous words in word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3514–3520, Online. https://doi.org/10.18653/v1/2020.emnlp-main.283

Loureiro, Daniel and Alípio Jorge. 2019a. Language modeling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence. https://doi.org/10.18653/v1/P19-1569

Loureiro, Daniel and Alípio Jorge. 2019b. LIAAD at SemDeep-5 challenge: Word-in-context (WiC). In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 1–5, Macau.

Luo, Fuli, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 2473–2482, Melbourne. https://doi.org/10.18653/v1/P18-1230,

Mallery, J. C. 1988. *Thinking about foreign policy: Finding an appropriate role for artificial intelligence computers*. Ph.D. Thesis, M.I.T. Political Science Department, Cambridge, MA.

Maru, Marco, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. SyntagNet: Challenging supervised word sense disambiguation with lexical-semantic combinations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3534–3540, Hong Kong. https://doi.org/10.18653/v1/D19-1359

McCarthy, Diana, Marianna Apidianaki, and Katrin Erk. 2016. Word sense clustering and clusterability. *Computational Linguistics*, 42(2):245–275. https://doi.org/10.1162/COLI_a_00247

McCrae, John Philip, Alexandre Rademaker, Ewa Rudnicka, and Francis Bond. 2020. English WordNet 2020: Improving and extending a WordNet for English using an open-source methodology. In *Proceedings of the LREC 2020 Workshop on Multimodal Wordnets (MMW2020)*, pages 14–19, Marseille.

McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Groberger. 2018. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861. https://doi.org/10.21105/joss.00861

Melamud, Oren, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin. https://doi.org/10.18653/v1/K16-1006

Mickus, Timothee, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2020. What do you mean, BERT? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 279–290, New York, NY.

Mihalcea, Rada, Timothy Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Lisbon.

Mihalcea, Rada and Andras Csomai. 2007. Wikify! Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference On Information And Knowledge Management*, pages 233–242, Lisbon. https://doi.org/10.1145/1321440.1321475

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, NV.

Miller, George A., Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, NJ. https://doi.org/10.3115/1075671.1075742

Moro, Andrea and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*. Denver, CO. https://doi.org/10.18653/v1/S15-2049

Moro, Andrea, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244. https://doi.org/10.1162/tacl_a_00179

Nair, Sathvik, Mahesh Srinivasan, and Stephan Meylan. 2020. Contextualized word embeddings encode aspects of human-like word sense knowledge. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 129–141, Online.

Navigli, Roberto. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69. https://doi.org/10.1145/1459352.1459355

Navigli, Roberto, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual word sense disambiguation. In *Proceedings of SemEval 2013*, pages 222–231, Atlanta, GA.

Navigli, Roberto and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250. https://doi.org/10.1016/j.artint.2012.07.001

Palmer, Martha, Hoa Dang, and Christiane Fellbaum. 2007. Making fine-grained and

coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163. https://doi.org/10.1017/S135132490500402X

Pasini, Tommaso. 2020. The knowledge acquisition bottleneck problem in multilingual word sense disambiguation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4936–4942. https://doi.org/10.24963/ijcai.2020/687

Pasini, Tommaso and Jose Camacho-Collados. 2020. A short survey on sense-annotated corpora. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5759–5765, Marseille.

Pasini, Tommaso and Roberto Navigli. 2018. Two knowledge-based methods for high-performance sense distribution learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5374–5381, New Orleans, LA.

Pasini, Tommaso and Roberto Navigli. 2020. Train-o-matic: Supervised word sense disambiguation with no (manual) effort. *Artificial Intelligence*, 279:103215. https://doi.org/10.1016/j.artint.2019.103215

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, LA. https://doi.org/10.18653/v1/N18-1202

Peters, Matthew, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels. https://doi.org/10.18653/v1/D18-1179

Peters, Matthew E., Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong. https://doi.org/10.18653/v1/D19-1005, PubMed: 31383442

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14. https://doi.org/10.18653/v1/W19-4302

Peterson, Daniel and Martha Palmer. 2018. Bayesian verb sense clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 5398–5405.

Pilehvar, Mohammad Taher and Jose Camacho-Collados. 2019. WiC: the word-in-context data set for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, MN.

Pilehvar, Mohammad Taher, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a seamless integration of word senses into downstream NLP applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1857–1869, Vancouver. https://doi.org/10.18653/v1/P17-1170

Pilehvar, Mohammad Taher and Roberto Navigli. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, 40(4):837–881. https://doi.org/10.1162/COLI_a_00202

Postma, Marten, Ruben Izquierdo, Eneko Agirre, German Rigau, and Piek Vossen. 2016. Addressing the MFS bias in WSD systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1695–1700, Portorož.

Postma, Marten, Ruben Izquierdo Bevia, and Piek Vossen. 2016. More is not always better: balancing sense distributions for all-words word sense disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3496–3506, Osaka.

Pradhan, Sameer, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of SemEval*, pages 87–92. https://doi.org/10.3115/1621474.1621490

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Raganato, Alessandro, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia. https://doi.org/10.18653/v1/E17-1010

Raganato, Alessandro, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167, Copenhagen. https://doi.org/10.18653/v1/D17-1120

Raganato, Alessandro, Tommaso Pasini, Jose Camacho-Collados, and Taher Mohammad Pilehvar. 2020. XL-WiC: A multilingual benchmark for evaluating semantic contextualization. *EMNLP 2020*, pages 7193–7206. https://doi.org/10.18653/v1/2020.emnlp-main.584

Reif, Emily, Ann Yuan, Martin Wattenberg, Fernanda B. Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems*, pages 8592–8600.

Reisinger, Joseph and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of ACL*, pages 109–117.

Resnik, Philip and David Yarowsky. 1999. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133. https://doi.org/10.1017/S1351324999002211

Rodd, Jennifer M. 2020. Settling into semantic space: An ambiguity-focused account of word-meaning access. *Perspectives on Psychological Science*, 15(2):411–427. https://doi.org/10.1177/1745691619885860, PubMed: 31961780

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866. https://doi.org/10.1162/tacl_a_00349

Rüd, Stefan, Massimiliano Ciaramita, Jens Müller, and Hinrich Schütze. 2011. Piggyback: Using search engines for robust cross-domain named entity recognition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 965–975, Portland, OR.

Saphra, Naomi and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267, Minneapolis, MN.

Scarlini, Bianca, Tommaso Pasini, and Roberto Navigli. 2019. Just "OneSeC" for producing multilingual sense-annotated data. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 699–709, Florence.

Scarlini, Bianca, Tommaso Pasini, and Roberto Navigli. 2020a. SensEmBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence*, pages 8758–8765. https://doi.org/10.1609/aaai.v34i05.6402

Scarlini, Bianca, Tommaso Pasini, and Roberto Navigli. 2020b. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539, Online. https://doi.org/10.18653/v1/2020.emnlp-main.285

Schütze, Hinrich. 1993. Word space. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 895–902, Morgan-Kaufmann.

Scozzafava, Federico, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. 2020. Personalized PageRank with syntagmatic information for multilingual word sense disambiguation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–46, Online. https://doi.org/10.18653/v1/2020.acl-demos.6

Severyn, Aliaksei, Massimo Nicosia, and Alessandro Moschitti. 2013. Learning semantic textual similarity with structural representations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 714–718, Sofia.

Shwartz, Vered and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419. https://doi.org/10.1162/tacl_a_00277

Soler, Aina Garí, Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. 2019. A comparison of context-sensitive models for lexical substitution. In *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*, pages 271–282.

Taghipour, Kaveh and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344, Beijing. https://doi.org/10.18653/v1/K15-1037

Taylor, Wilson L. 1953. "Cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433. https://doi.org/10.1177/107769905303000401

Tenney, Ian, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence. https://doi.org/10.18653/v1/P19-1452

Tenney, Ian, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *Proceeding of the 7th International Conference on Learning Representations (ICLR)*.

Usbeck, Ricardo, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, and others. 2015. GERBIL: General entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1133–1143.

van Schijndel, Marten, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5830–5836, Hong Kong. https://doi.org/10.18653/v1/D19-1592

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Vial, Loïc, Benjamin Lecouteux, and Didier Schwab. 2018. UFSAC: Unification of sense annotated corpora and tools. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki.

Vial, Loïc, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic state-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Voita, Elena, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the Transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4396–4406, Hong Kong.

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors. *Advances in Neural Information Processing Systems*, volume 32, Curran Associates, Inc.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels. https://www.aclweb.org/anthology/W18-5446.

Wiedemann, Gregor, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does

BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, and others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Yaghoobzadeh, Yadollah, Katharina Kann, T. J. Hazen, Eneko Agirre, and Hinrich Schütze. 2019. Probing for semantic classes: Diagnosing the meaning content of word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5740–5753, Florence. https://doi.org/10.18653/v1/P19-1574

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763, Curran Associates, Inc.

Yenicelik, David, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT capture semantics? A closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162, Online. https://doi.org/10.18653/v1/2020.blackboxnlp-1.15

Yuan, Dayu, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1374–1385, Osaka.

Zhong, Zhi and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL System Demonstrations*, pages 78–83, Uppsala.

Zhou, Wangchunshu, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence. https://doi.org/10.18653/v1/P19-1328

Zipf, George K. 1949. *Human behavior and the Principle of Least-Effort*, Addison-Wesley, Cambridge, MA.

Appendix F

On the Cross-lingual Transferability of Contextualized Sense Embeddings

Submitted: August 2021; Published: November 2021.

Kiamehr Rezaee†, Daniel Loureiro†, Mohammad Taher Pilehvar, Jose Camacho-Collados († equal contribution). 2021. In Proceedings of the 1st Workshop on Multilingual Representation Learning, pages 107–115, Punta Cana, Dominican Republic. Association for Computational Linguistics.
Published PDF: http://dx.doi.org/10.18653/v1/2021.mrl-1.10.

Relevant Contributions

- Demonstrates feasibility of learning sense embeddings from multilingual NLMs (i.e., XLM-R [132]) that can be used for WSD in various languages (English, French, German, Spanish, Italian, and Farsi – limited by available annotations and test sets).

Return to Table of Contents

On the Cross-lingual Transferability of Contextualized Sense Embeddings

Kiamehr Rezaee[1*], **Daniel Loureiro**[2*], **Jose Camacho-Collados**[3], **Mohammad Taher Pilehvar**[4]
[1] Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[2] LIAAD - INESC TEC, Department of Computer Science - FCUP, University of Porto, Portugal
[3] School of Computer Science and Informatics, Cardiff University, United Kingdom
[4] Tehran Institute for Advanced Studies, Khatam University, Tehran, Iran
k_rezaee@comp.iust.ac.ir, daniel.b.loureiro@inesctec.pt,
camachocolladosj@cardiff.ac.uk, mp792@cam.ac.uk

Abstract

In this paper we analyze the extent to which *contextualized sense embeddings*, i.e., sense embeddings that are computed based on contextualized word embeddings, are transferable across languages. To this end, we compiled a unified cross-lingual benchmark for Word Sense Disambiguation. We then propose two simple strategies to transfer sense-specific knowledge across languages and test them on the benchmark. Experimental results show that this contextualized knowledge can be effectively transferred to similar languages through pre-trained multilingual language models, to the extent that they can outperform monolingual representations learned from existing language-specific data.

1 Introduction

Word Sense Disambiguation (WSD) is an indispensable component of language understanding (Navigli, 2009); hence, it has been one of the most studied long-standing problems in lexical semantics. Currently, the dominant WSD paradigm is the supervised approach (Raganato et al., 2017), which highly relies on sense-annotated data. Similarly to many other supervised tasks, the amount of labeled (sense-annotated) data for WSD highly determines downstream performance. One of the factors that make WSD a challenging problem is that creating sense-annotated data is an expensive and arduous process, i.e., the so-called knowledge-acquisition bottleneck (Gale et al., 1992). Moreover, WSD research often focuses on the English language. While datasets for other languages exist (Petrolito and Bond, 2014a; Pasini and Camacho-Collados, 2020), these are generally automatically generated (Delli Bovi et al., 2017; Pasini et al., 2018; Scarlini et al., 2020a; Barba et al., 2020) or not large enough for training supervised WSD models (Navigli et al., 2013a; Moro and Navigli, 2015).[1]

However, recent contextualized embeddings have proven highly effective in English WSD (Peters et al., 2018; Loureiro and Jorge, 2019; Vial et al., 2019; Loureiro et al., 2021), as well as in capturing high-level linguistic knowledge that can be shared or transferred across different languages (Conneau et al., 2020; Cao et al., 2020). Therefore, cross-lingual transfer has opened new opportunities to circumvent the knowledge acquisition bottleneck for less-resourced languages. In this paper, we aim at investigating this opportunity. To this end, we build upon recent research on cross-lingual transfer to compute contextualized sense embeddings and verify if semantic distinctions in the English language are transferable to other languages.

The contributions are threefold: (1) We adapt existing datasets to build a unified benchmark for cross-lingual WSD based on WordNet; (2) we test the effectiveness of contextualized embeddings for cross-lingual transfer in the context of WSD; and (3) we establish relevant and simple baselines for future work in cross-lingual WSD.[2]

2 Related Work

This works lies at the intersection of two areas of NLP research: Word Sense Disambiguation and cross-lingual semantic representation. Hence, we cover the recent relevant work in the corresponding literature.

2.1 Word Sense Disambiguation

WSD techniques can be broadly put into two categories: knowledge-based and supervised. The main difference lies in that the latter makes use of sense-annotated data for its training phase, whereas the former exploits the encoded knowledge in sense in-

Authors marked with a star (*) contributed equally.

[1] An exception to this pattern is the recent XL-WSD benchmark (Pasini et al., 2021), contemporary to this paper.

[2] Data and code are available at https://github.com/danlou/Zero-MWSD

Proceedings of the 1st Workshop on Multilingual Representation Learning, pages 107–115
November 11, 2021.

	Train - SemCor		Test - SemEval-15			Test - SemEval-13					Test - FN
	EN	IT	EN	IT	ES	EN	FR	DE	ES	IT	FA
Nouns	87,002	43,058	512	515	512	1,637	1,438	958	1,176	1,448	3,063
Verbs	88,334	25,164	252	233	260	–	–	–	–	–	29
Adj.	31,753	16,029	136	159	119	–	–	–	–	–	366
Adv.	18,947	7,951	82	25	53	–	–	–	–	–	40
ALL	226,036	92,202	982	932	944	1,637	1,438	958	1,176	1,448	3,498
RAW	226,036	92,202	1,175	1,151	1,155	1,931	1,656	1,467	1,481	1,706	4,272

Table 1: Number of sense-annotated instances in the benchmark datasets after cleaning and unification. RAW counts correspond to number of instances in the original datasets before cleaning and unification.

ventories such as WordNet (e.g. semantic relations, sense glosses, distributions, etc.) for inference.

For the last decade, the supervised approach has been the dominant paradigm for WSD (Raganato et al., 2017), either the conventional feature-based systems (Zhong and Ng, 2010; Iacobacci et al., 2016), LSTM-driven techniques (Melamud et al., 2016; Yuan et al., 2016), or the more recent trend empowered by pre-trained language models (Loureiro and Jorge, 2019; Scarlini et al., 2020b). In the latter approaches, feature extraction strategies where sense embeddings are determined by averaging a word's contextualised representations have proven surprisingly effective (Loureiro et al., 2021), even in multilingual settings (Bevilacqua and Navigli, 2020; Raganato et al., 2020). We extend this simple idea to the cross-lingual setting, showing that the vanilla contextualized sense embeddings achieving outstanding results in the monolingual setting can also be effective for transferring knowledge across languages.

2.2 Cross-lingual representation

WSD performance is largely dependent on the availability of large amounts of manually-curated sense annotations. However, as usual in NLP, most of the sense-annotated corpora are dedicated to the English language only. Nonetheless, recent work on cross-lingual word embeddings has shown that it is possible to reliably align monolingual semantic spaces with minimal or no supervision (Artetxe et al., 2017; Zhang et al., 2017; Conneau et al., 2018). Moreover, pre-trained language models, like BERT (Devlin et al., 2019), have been shown to be effective in transferring knowledge across languages (Lample and Conneau, 2019; Pires et al., 2019; Artetxe et al., 2020). In this paper we build on these ideas to take the best of both worlds. Instead of transferring static word embeddings, the main idea is to learn sense embeddings (learned using multilingual language models) that can be shared across languages.

3 Cross-lingual WSD Benchmark

In order to develop a unified benchmark for cross-lingual Word Sense Disambiguation, we opted for Princeton **WordNet** (Fellbaum, 1998, PWN) as our reference sense inventory. Thanks to its completeness (covering different parts of speech) and open nature, PWN is regarded as the de facto sense inventory for WSD in English. Moreover, the multilingual efforts from Open Multilingual WordNet (Bond and Foster, 2013), linked to the English PWN, make this resource prompt to extensions for sense-annotated corpora in other languages. We use PWN v3.0 for all our experiments, including the unified cross-lingual benchmark, converting all datasets to both the same XML schema of Raganato et al. (2017) and a practical *json* format. In the following we describe the datasets used to build the cross-lingual WSD benchmark, with Table 1 summarizing their main statistics.

3.1 SemCor training sets

As training corpora we used SemCor (Miller et al., 1993), which consists of a collection of English documents annotated with PWN senses. Despite its age, SemCor remains the standard training corpus for WSD due to its large number of manual sense annotations. There have been several efforts towards providing sense annotations for translated versions of SemCor (Petrolito and Bond, 2014a). Consequently, we also considered the Italian version of SemCor included in MultiSemCor (Bentivogli and Pianta, 2005), which is the language with most PWN annotations available. MultiSemCor includes sense annotations from PWN v1.6; hence, we used the mappings from Daudé et al. (2000) to convert these into PWN v3.0 annotations.

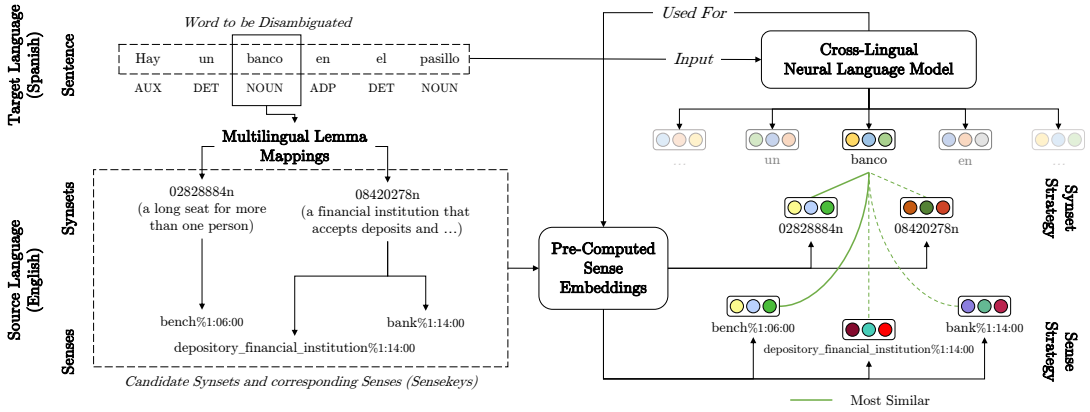


Figure 1: Overview of the proposed method for multilingual zero-shot word sense disambiguation. The example sentence presented ('There is a bench in the hall', in English) is using a different language (target) than our sense inventory (source), but using multilingual language models and lemma mappings, we demonstrate how it's still possible to perform disambiguation, using either variations of our method (sense and synset strategies).

3.2 Multilingual SemEval test sets

We considered two multilingual datasets: SemEval 2013 (Navigli et al., 2013a) available for English, French and Italian, and SemEval 2015 (Moro and Navigli, 2015), available for English, French, Italian, Spanish and German. These datasets were annotated with BabelNet (Navigli and Ponzetto, 2012), a resource that contains WordNet, among other linked sense inventories. Therefore, from each dataset we simply considered those disambiguated instances that could be mapped to PWN 3.0, while the rest of instances were removed. We also rely on BabelNet to gather a representative set of candidate senses for any given target word.

3.3 FarsNet

To extend the evaluation set beyond European languages, we first performed an exhaustive search for available WSD datasets that could be integrated into our benchmark, unsuccessfully.[3] To fill this gap, we constructed an evaluation set for a distant low-resource language: Farsi. This dataset was constructed based on example sentences provided with the FarsNet project (Shamsfard et al., 2010). As the largest Farsi WordNet available, FarsNet is constructed in a semi-automatic manner. In its latest version (v3.0), the lexical resource covers more than 100K lexical entries in around 40K synsets. FarsNet synsets are aligned with those in PWN 3.0, whenever a link could be established. This allows utilizing the resource in cross-lingual applications.

Many of the synsets in FarsNet are provided with a usage example sentence for one of the terms in the corresponding Farsi synset. We take this as the basis for the construction of the Farsi dataset. Specifically, there are more than 30K Farsi usage examples that are linked to the corresponding synsets in PWN. From these, we extract a set of 4,272 sentences for 3,498 unique target words, after discarding monosemous words and filtering. The distribution of instances over the four parts of speech can be found in Table 1.

4 Methodology

We describe two WSD strategies based on contextualized embeddings (§4.1) and propose adaptations to the cross-lingual setting (§4.2). Figure 1 provides an overview of the overarching methodology.

4.1 Contextualized Embeddings for WSD

One of the most effective, yet simple, solutions for WSD is matching contextualized embeddings against precomputed sense embeddings learned using the same Neural Language Model (NLM). This approach has been used by earlier works (Melamud et al., 2016; Peters et al., 2018; Loureiro and Jorge, 2019), achieving state-of-the-art results. In these works, sense embeddings (or what we refer to as *contextualized sense embeddings*) are computed from the average of all corresponding contextual embeddings in sense-annotated corpora. One limi-

[3] Our active search efforts are described in the appendix. In general, existing WSD datasets were either under a restrictive license, not available anymore, or not linkable to English PWN.

tation of contextualized sense embeddings is that they only cover those senses that are present in the underlying training corpus. This issue can be alleviated by exploiting the structure of the semantic network. For this, we leverage the simple graph-based propagation method described in Loureiro and Jorge (2019), which allows for a coverage of the entire sense inventory of PWN.

With this strategy we can easily test whether contextualized sense embeddings can be transferred across languages with a solution purely based on matching nearest neighbors only, without any other artifacts. Below we describe the sense-based strategy generally used in the literature for monolingual WSD (sense strategy), and propose a new strategy that can be directly used in a cross-lingual setting (synset strategy).

Sense strategy. This strategy is the standard approach used in most WSD methods. After computing our sense embeddings from sense-annotated corpora, we disambiguate target words during testing based on a nearest neighbour strategy using their contextualized embeddings.

Synset strategy. In this alternative strategy, we learn synset embeddings by converting the annotated sense labels to synset labels, thus learning representations from multiple word senses that refer to the same concept, and becoming less reliant on lexical features.

4.2 Cross-lingual Adaptations

In cross-lingual experiments we are given sense-annotated corpora in a source language but not in the target language. Therefore, a multilingual NLM such as mBERT (Devlin et al., 2019) or XLM-R (Lample and Conneau, 2019) is required.

We adapt the sense strategy to the cross-lingual setting as follows. Given the lemma and part-of-speech of a word in the target language, we first gather all the candidate synsets in the source language from Babelnet. Then each candidate synset is associated with one or more senses. For example, we can find two candidate PWN senses for the word *presente* (*present* in Spanish). The first sense corresponds to the PWN synset "intermediate between past and future" and the second to "being or existing in a specified place". Finally, we compute the cosine distance from the contextualized embeddings of target word (*presente*) to all the candidate contextualized sense embeddings from the source language (*present%3:00:01::* and *present%3:00:02::*, respectively), and select the closest candidate sense. Note that the synset strategy does not require adaptation because synsets are language-independent.

5 Evaluation

We evaluate the proposed strategies in two different settings, using WSD as our test bed: monolingual (Section 5.1) and cross-lingual (Section 5.2).

Experimental setting. We use **RoBERTa** (Liu et al., 2019) for monolingual experiments, and **XLM-R** (Conneau et al., 2020) for cross-lingual experiments.[4] These two models are state-of-the-art for English and multilingual tasks while sharing a very similar architecture. The most important difference between these models is that while RoBERTa is pre-trained only on English texts, XLM-R is pre-trained on 100 languages (unevenly distributed). We use the large variants of these models (355M parameters for each).[5] All results are measured according to the F-measure.

5.1 Experiment 1: Monolingual WSD

Before delving into the cross-lingual experiments, we present monolingual results in English and Italian (languages with training data) in Table 2. The aim of this experiment is twofold: (1) compare the effectiveness of the disambiguation strategies, and (2) compare the performance of monolingual (RoBERTa) and multilingual models (XLM-R).

Baselines. As additional baselines, we add the results of the original BERT-based **LMMS** model (Loureiro and Jorge, 2019) and Context2Vec (**C2V**) (Melamud et al., 2016), which is also based on a simple nearest neighbors strategy, in this case with an LSTM instead of a transformer model.

Results. Table 2 shows the results of this English monolingual experiment.[6] We report results in all datasets from the unified WSD evaluation framework of Raganato et al. (2017).[7] As expected, the

[4] Following Loureiro and Jorge (2019), we consider token-level embeddings as the average of sub-token embeddings, which is computed as the sum of embeddings from the last 4 layers of the corresponding NLM.

[5] Our code is based on the Fairseq toolkit (Ott et al., 2019). We run our experiments on a single RTX 2070, with a runtime under 2 hours for generating all embeddings used in this work.

[6] We experimented with NLMs trained exclusively on Italian (i.e. UmBERTo-CC and dbmdz-IT-XXL), but found that the senses learned using those models do not consistently outperform the MFS baseline on both test sets.

[7] Datasets of the unified WSD framework: Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Mihalcea et al., 2004),

Model	Strategy	SensEval-2	SensEval-3	SemEval-2007	SemEval-2013	SemEval-2015	ALL
RoBERTa	Sense	**75.5**	73.5	**69.2**	72.2	75.9	**73.9**
	Synset	74.4	**74.1**	68.4	72.1	**76.0**	73.6
XLM-R	Sense	71.0	67.8	61.5	70.1	72.1	69.5
	Synset	70.0	67.2	60.9	69.7	72.3	69.0
LMMS	Sense	*75.4*	*74.0*	*66.4*	***72.7***	*75.3*	*73.8*
C2V	Sense	*71.8*	*69.1*	*61.3*	*65.6*	*71.9*	*69.0*
MFS	–	*65.6*	*66.0*	*54.5*	*63.8*	*67.1*	*64.8*

Table 2: English monolingual F1 results on the evaluation framework of Raganato et al. (2017) for the two strategies: Sense (Se) and Synset (Sy).

purely monolingual model performs better than the multilingual one. As for the strategies, the usual sense strategy shows better performance. Nonetheless, the language-independent synset strategy attains competitive results, clearly outperforming a strong baseline such as Context2Vec, for example.

5.2 Experiment 2: Cross-lingual Transfer

We experimented with a pure zero-shot cross-lingual setting where senses are learned in one language (in our case English or Italian) and directly evaluated on another language. We employ the two strategies explained in Section 4.

Baselines. As baselines we include a random baseline (i.e., randomly picking a sense/synset from the target language's inventory), and a system that relies on the **static** word embeddings from XLM-R (i.e., input layer embeddings of XLM-R without making use of the context), instead of the **contextualized** sense embeddings obtained as described in Section 4.1.[8] The reason behind the possibility of using static word embeddings in this WSD setting lies in the fact that different senses of a word may be translated into different words in another language. This baseline makes use of the same sense and synset strategies.

Results. Table 3 shows the results for the zero-shot cross-lingual WSD experiment.[9] XLM-R outperforms the baselines by a large margin and proves to be robust in the cross-lingual setting (with performance in the same ballpark as in the monolingual setting). In particular, the fact that XLM-R outperforms the static embedding baseline by a large margin reinforces the idea that contextualized sense embedding are indeed transferable across languages (at least similar ones) to a large extent, which in turn opens up interesting avenues for future work on cross-lingual WSD. Nonetheless, as with English WSD, there are still many open questions as to what extent the fine granularity of PWN can be captured by automatic models.

Finally, as expected, learning representations using the larger English SemCor provides consistently better results than the smaller Italian counterpart, except for the distant language Farsi. More interestingly, XLM-R senses learned from English data can outperform the senses from the same model learned from language-specific data in the Italian test sets. Nonetheless, the simple synset strategy on Italian clearly surpasses the static baselines as well.

6 Conclusions

In this paper, we analyzed to what extent contextualized embeddings can be transferred across languages, using WSD as our test bed. To this end, we developed a unified framework that can be used for evaluating cross-lingual models. The first results are encouraging, as they show that multilingual language models can learn contextualized sense embeddings that can be effectively transferred from one language to another, attaining competitive results in WSD with no access to annotated data in the target language or external resources. One limitation of this work is in the nature of languages evaluated, which are all Indo-European for which test data was available. As future work it will be interesting to extend this benchmark to languages

SemEval-2007 (Agirre et al., 2007), SemEval-2013 (Navigli et al., 2013b), and SemEval-2015 (Moro and Navigli, 2015).

[8] We also tried to include Babelfy (Moro et al., 2014) as a multilingual knowledge-based baseline, without success. The latest public API of Babelfy misses over 50% of the instances in our benchmark and therefore the recall was suboptimal.

[9] English monolingual results slightly differ from those in Table 2, as in this case we focused on the BabelNet portion of the SemEval datasets (as explained in Section 3.2). For Italian we only report the synset strategy, as we could not have access to all the senses in MultiWordNet (Pianta et al., 2002).

Language	Strategy	Type	SemEval-13					SemEval-15			FN
			EN	FR	DE	ES	IT	EN	IT	ES	FA
English	**Se**	Static	*38.1*	28.0	50.1	38.7	41.0	*41.5*	45.2	40.6	48.7
		Contextualized	***67.4***	**60.7**	57.5	**69.7**	**66.1**	***71.7***	69.1	**68.0**	55.4
	Sy	Static	*41.0*	30.6	48.2	39.4	43.2	*39.8*	46.2	43.6	48.2
		Contextualized	*66.3*	59.0	**58.3**	66.8	64.9	*71.4*	**69.5**	67.2	**56.4**
Italian	**Sy**	Static	51.3	41.0	55.1	54.4	48.3	54.7	54.6	56.8	46.7
		Contextualized	63.2	55.9	55.4	65.9	*62.9*	67.1	*67.2*	65.4	**56.4**
Random baseline			37.8	25.8	47.9	38.7	38.6	41.7	44.5	38.0	49.5

Table 3: F1 results using zero-shot cross-lingual transfer (XLM-R) with English or Italian annotations. Two different types of sense embedding: static (S) and contextualized (C). Monolingual setting (i.e., learn and test in the same language) is also included for completeness.

from different families, for which cross-lingual embedding transfer has been shown to be more challenging (Glavaš et al., 2019; Doval et al., 2020).

References

Eneko Agirre, Lluís Màrquez, and Richard Wicentowski, editors. 2007. *Proceedings of SemEval-2007*.

Ali Alkhatlan, Jugal Kalita, and Ahmed Alhaddad. 2018. Word sense disambiguation for arabic exploiting arabic wordnet and word embedding. *Procedia Computer Science*, 142:50 – 60. Arabic Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations.

Edoardo Barba, Luigi Procopio, Niccolo Campolungo, Tommaso Pasini, and Roberto Navigli. 2020. Mulan: Multilingual label propagation for word sense disambiguation. In *Proc. of IJCAI*, pages 3837–3844.

Luisa Bentivogli and Emanuele Pianta. 2005. Exploiting parallel texts in the creation of multilingual semantically annotated resources: the multisemcor corpus. *Natural Language Engineering*, 11(3):247–261.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria. Association for Computational Linguistics.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *Proceedings of ICLR*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of ICLR*.

J. Daudé, L. Padró, and G. Rigau. 2000. Mapping WordNets using structural information. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 504–511, Hong Kong. Association for Computational Linguistics.

Claudio Delli Bovi, Jose Camacho-Collados, Alessandro Raganato, and Roberto Navigli. 2017. EuroSense: Automatic harvesting of multilingual sense annotations from parallel text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 594–600, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Mona Diab, Musa Alkhalifa, Sabry ElKateb, Christiane Fellbaum, Aous Mansouri, and Martha Palmer. 2007. SemEval-2007 task 18: Arabic semantic labeling. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 93–98, Prague, Czech Republic. Association for Computational Linguistics.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa Anke, and Steven Schockaert. 2020. On the robustness of unsupervised and semi-supervised cross-lingual word embedding learning. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4013–4023, Marseille, France. European Language Resources Association.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

William A. Gale, Kenneth Church, and David Yarowsky. 1992. A method for disambiguating word senses in a corpus. *Computers and the Humanities*, 26:415–439.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721, Florence, Italy. Association for Computational Linguistics.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany. Association for Computational Linguistics.

Peng Jin, Yunfang Wu, and Shiwen Yu. 2007. SemEval-2007 task 05: Multilingual Chinese-English lexical sample. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 19–23, Prague, Czech Republic. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, pages 1–55.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28.

Rada Mihalcea and Phil Edmonds. 2004. Senseval-3: Third international workshop on the evaluation of systems for the semantic analysis of text. In *Association for Computational Linguistics*, volume 4.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, N.J.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244.

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013a. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013b. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Proceedings of SemEval 2013*, pages 222–231.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono. 2010. SemEval-2010 task: Japanese WSD. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 69–74, Uppsala, Sweden. Association for Computational Linguistics.

Zeynep Orhan, Emine Çelik, and Demirgüç Neslihan. 2007. SemEval-2007 task 12: Turkish lexical sample task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 59–63, Prague, Czech Republic. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Tommaso Pasini and Jose Camacho-Collados. 2020. A short survey on sense-annotated corpora. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5759–5765, Marseille, France. European Language Resources Association.

Tommaso Pasini, Francesco Elia, and Roberto Navigli. 2018. Huge automatically extracted training-sets for multilingual word SenseDisambiguation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tommaso Pasini, Alessandro Raganato, Roberto Navigli, et al. 2021. Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Tommaso Petrolito and Francis Bond. 2014a. A survey of WordNet annotated corpora. In *Proceedings of the Seventh Global Wordnet Conference*, pages 236–245, Tartu, Estonia. University of Tartu Press.

Tommaso Petrolito and Francis Bond. 2014b. A survey of wordnet annotated corpora. In *Proceedings Global WordNet Conference, GWC-2014*, pages 236–245.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *First international conference on global WordNet*, pages 293–302.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of EACL*, pages 99–110, Valencia, Spain.

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A multilingual benchmark for evaluating semantic contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206, Online. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020a. Sense-annotated corpora for word sense disambiguation in multiple languages and domains. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5905–5911.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020b. SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence.

Mehrnoush Shamsfard, Akbar Hesabi, Hakimeh Fadaei, Niloofar Mansoory, Payam Noor, Ali Reza Gholi Famian, Somayeh Bagherbeigi, Elham Fekri, and Maliheh Monshizadeh. 2010. Semi automatic development of FarsNet; the Persian WordNet. In *Proceedings of 5th global WordNet conference*.

Kiyoaki Shirai. 2002. Construction of a word sense tagged corpus for SENSEVAL-2 Japanese dictionary task. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. In *Proceedings of the 10th Global WordNet Conference*.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING*, pages 1374–1385.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945, Copenhagen, Denmark. Association for Computational Linguistics.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.

A Appendix: Compilation of WordNet-based WSD Datasets

In addition to the datasets included in our benchmark, mostly composed of European languages, we made an effort to retrieve and compile datasets for other languages. In Table 4 we provide details about our unsuccessful attempts and issues to integrate existing WSD datasets in the literature, mainly taken from DKPro[10] and the survey of Petrolito and Bond (2014b). Not only are most of these datasets unavailable, we also learn from their respective publications that the sense inventories used often aren't based on WordNet, and thus would require manual remapping of annotations for integration into our benchmark.

[10] https://dkpro.github.io/dkpro-wsd/corpora/

Resource	Language	# Instances	Inventory	Availability	License
Senseval-2 (Shirai, 2002)	Japanese	10,000	IKJ	N/A	N/A
Senseval-2 (Edmonds and Cotton, 2001)	Korean	N/A	N/A	N/A	N/A
Senseval-3 (Mihalcea and Edmonds, 2004)	Chinese	1,204	HowNet	Publicly Avail.	Public Domain
SemEval-2007 Task 5 (Jin et al., 2007)	Chinese	3,621	CSD	N/A	N/A
SemEval-2007 Task 11 (Orhan et al., 2007)	Turkish	5,385	TKD	N/A	N/A
SemEval-2007 Task 18 (Diab et al., 2007)	Arabic	888	AWN	N/A	N/A
SemEval-2010 Task 16 (Okumura et al., 2010)	Japanese	2,500	IKJ	On Request	Restrictive
OntoNotes (Weischedel et al., 2011)	Arabic	200K	Coarse WN	For Members	Restrictive
OntoNotes (Weischedel et al., 2011)	Chinese	800K	Coarse WN	For Members	Restrictive
Alkhatlan et al. (2018)	Arabic	240	AWN	N/A	N/A

Table 4: Details about the various WSD datasets covering non-European languages surveyed in our work (N/A: Not Available; WN: WordNet; AWN: Arabic WordNet; we refer to respective papers for remaining acronyms).

Appendix G

LMMS Reloaded: Transformer-based Sense Embeddings for Disambiguation and Beyond

Submitted: May 2021; Published: January 2022; SJR: Q1.

Daniel Loureiro and Alípio Jorge and José Camacho-Collados. 2022. In Artificial Intelligence, vol. 305, p. 103661. Published PDF: https://doi.org/10.1016/j.artint.2022.103661.

Relevant Contributions

- Proposes a principled approach for weighted layer pooling towards improved sense representations.
- Reports extensive evaluation of sense embeddings across six tasks, and numerous ablation analyses targeting main choices of our methods.
- Generalizes the full LMMS approach for additional NLMs (i.e., RoBERTa, XLNet and ALBERT).
- Presents surprising findings regarding layerwise performance for sense-related tasks, and NLMs specially suited for particular task subsets.

Return to Table of Contents

LMMS Reloaded: Transformer-based Sense Embeddings for Disambiguation and Beyond

Daniel Loureiro[a,*], Alípio Mário Jorge[a], Jose Camacho-Collados[b]

[a]*LIAAD - INESC TEC, Dept. of Computer Science, FCUP, University of Porto, Portugal*
[b]*School of Computer Science and Informatics, Cardiff University, United Kingdom*

Abstract

Distributional semantics based on neural approaches is a cornerstone of Natural Language Processing, with surprising connections to human meaning representation as well. Recent Transformer-based Language Models have proven capable of producing contextual word representations that reliably convey sense-specific information, simply as a product of self-supervision. Prior work has shown that these contextual representations can be used to accurately represent large sense inventories as sense embeddings, to the extent that a distance-based solution to Word Sense Disambiguation (WSD) tasks outperforms models trained specifically for the task. Still, there remains much to understand on how to use these Neural Language Models (NLMs) to produce sense embeddings that can better harness each NLM's meaning representation abilities. In this work we introduce a more principled approach to leverage information from all layers of NLMs, informed by a probing analysis on 14 NLM variants. We also emphasize the versatility of these sense embeddings in contrast to task-specific models, applying them on several sense-related tasks, besides WSD, while demonstrating improved performance using our proposed approach over prior work focused on sense embeddings. Finally, we discuss unexpected findings regarding layer and model performance variations, and potential applications for downstream tasks.

*Corresponding author
**AIJ Publication: https://doi.org/10.1016/j.artint.2022.103661
Email addresses: daniel.b.loureiro@inesctec.pt (Daniel Loureiro), amjorge@fc.up.pt (Alípio Mário Jorge), camachocolladosj@cardiff.ac.uk (Jose Camacho-Collados)

1. Introduction

Lexical ambiguity is prevalent across different languages and plays an important role in improving communication efficiency (Piantadosi et al., 2012). Word Sense Disambiguation (WSD) is a long-standing challenge in the field of Natural Language Processing (NLP), and Artificial Intelligence more generally, with an extended history of research in computational linguistics (Navigli, 2009).

Interestingly, both computational and psychological accounts of meaning representation have converged on high-dimensional vectors within semantic spaces.

From the computational perspective, there is a rich line of work on learning word embeddings based on statistical regularities from unlabeled corpora, following the well-established Distributional Hypothesis (Harris, 1954; Firth, 1957, DH). The first type of distributional word representations relied on count-based methods, initially popularized by LSA (Deerwester et al., 1990), and later refined with GloVe (Pennington et al., 2014). Before GloVe, word embeddings learned with neural networks, first introduced by Bengio et al. (2003a), gained wide adoption with word2vec (Mikolov et al., 2013a) and, afterwards, culminated with fastText (Bojanowski et al., 2017). The development and improvement of word embeddings has been a major contributor to the progress of NLP in the last decade (Goldberg, 2017).

From the psychological perspective, there is also ample behavioural evidence in support of distributional representations of word meaning. Similarly to word embeddings, these representations are related according to the degree of shared features within semantic spaces, which translates into proximity in vector-space (Rodd, 2020; Klein & Murphy, 2001). Understandably, the nature of the features making up this psychological account of semantic space, among other aspects (e.g., learning method), is not as clear as we find in the computational account. Nevertheless, contextual co-occurrence is among the most informative factors for meaning representation as well (Mcdonald & Ramscar, 2001; Erk, 2016; Radach et al., 2017). There are even use cases in neurobiology motivating research into accurate distributional representations of word meaning. In Pereira et al.

(2018), word embeddings have proven useful for decoding words and sentences from brain activity, after learning a mapping between corpus-based embeddings (i.e., GloVe and word2vec) and fMRI activation.

The current understanding of how humans perform disambiguation attributes major relevance to sentential context, and other linguistic and paralinguistic cue's (e.g., speaker accent) to a lesser extent (Rodd, 2020; Cai et al., 2017). However, the previously mentioned computational approaches are not designed for sense-level representation due to the Meaning Conflation Deficiency (Camacho-Collados & Pilehvar, 2018), as they converge different senses into the same word-level representation. Some works have explored variations on the word2vec method for sense-level embeddings (Rothe & Schütze, 2015; Iacobacci et al., 2015; Pilehvar & Collier, 2016; Mancini et al., 2017), but the dynamic word-level interactions composing sentential context were not targeted by those works.

The works of Melamud et al. (2016); Yuan et al. (2016); Peters et al. (2018a) were among the first to propose Neural Language Models (NLMs) featuring dynamic word embeddings conditioned on sentential context (i.e., contextual embeddings). These works showed that NLMs (trained exclusively on language modelling objectives) can produce contextual embeddings for word forms that are sensitive to the word's usage in particular sentences. Furthermore, these works also addressed WSD tasks with a simple nearest neighbours solution (k-NN) based on proximity between contextual embeddings. Their results rivalled systems trained specifically for WSD (i.e., with additional modelling objectives), highlighting the accuracy of these contextual embeddings.

However, it was not until the development of Transformer-based NLMs, namely BERT (Devlin et al., 2019), that contextual embeddings from NLMs showed clearly better performance on WSD tasks than previous systems trained specifically for WSD (LMMS, Loureiro & Jorge, 2019a).

In this earlier work, we explored how to further take advantage of the representational power of NLMs through propagation strategies and encoding sense definitions. Besides pushing the state-of-the-art of WSD, in Loureiro & Jorge (2019a) we created sense embeddings for every entry in the Princeton Word-

Net v3.0 (200k word senses, Fellbaum, 1998), so that the semantic space being represented is granular and expansive enough to encompass general knowledge domains for various parts-of-speech of the English language. With this fully populated semantic space at our disposal we suggested strategies for uncovering biases and world knowledge represented by NLMs.

Since our work on LMMS, others have shown additional performance gains for WSD with fine-tuning or classification approaches that make better usage of sense definitions (Huang et al., 2019; Blevins & Zettlemoyer, 2020), semantic relations from external resources (Scarlini et al., 2020a; Bevilacqua & Navigli, 2020), or altogether different approaches to WSD (Barba et al., 2021).

However, there are several questions still standing regarding how to leverage NLMs for creating accurate and versatile sense embeddings, beyond optimizing for WSD benchmarks only. Given that semantic spaces with distributional representations of word meanings feature prominently in both the conventional computational and psychological accounts of word disambiguation, these questions warrant further exploration.

Contributions. In this extension of LMMS, we broaden our scope to more recent Transformer-based models in addition to BERT (Yang et al., 2019; Liu et al., 2019b; Lan et al., 2020) (14 model variants in total), verify whether they exhibit similar proficiency at sense representation, and explore how performance variation can be attributed to particular differences in these models. Striving for a principled approach to sense representation with NLMs, we also introduce a new layer pooling method, inspired by recent findings of layer specialization (Reif et al., 2019), which we show is crucial to effectively use these new NLMs for sense representation. Most importantly, in this article we provide a general framework for learning sense embeddings with Transformers and perform an extensive evaluation of such sense embeddings from different NLMs on various sense-related tasks, emphasizing the versatility of these representations.

Outline. This work is organized as follows. We first provide some background information on the main topics of this research: Vector Semantics (§2.1), Neural Language Modelling (§2.2) and Sense Inventories (§2.3). Next, we describe related work on Sense Embeddings (§3.1), WSD (§3.2) and Probing NLMs (§3.3).

The method used to produce this work's sense embeddings is described in Section 4, covering aspects of the method introduced in Loureiro & Jorge (2019a) (from §4.1 to §4.3), as well as our new layer pooling method in Section 4.4.

In Section 5 we describe our experimental setting, providing relevant details about our choice of NLMs (§5.1) and annotated corpora used to learn sense embeddings (§5.2).

The layer pooling methodology described in Section 4.4 requires validating performance under two distinct modes of application. Consequently, in Section 6 we report on performance variation per layer across NLMs (§6.1), highlight differences between disambiguation and matching profiles (§6.2), and present the rationale for choosing particular profiles for each task (§6.3).

In Section 7, we tackle several sense-related tasks using our proposed sense embeddings and compare results against the state-of-the-art, namely: WSD (§7.1), Uninformed Sense Matching (§7.2), Word-in-Context (§7.3), Graded Word Similarity in Context (§7.4) and Paired Sense Similarity (§7.5).

In order to better understand the contributions of this work, Section 8 reports on several ablation analyses targeting the following: choice of Sense Profiles (§8.1), impact of unambiguous word annotations (§8.2), merging gloss representations (§8.3), and indirect representation of synsets (§8.4).

We discuss our findings in Section 9, regarding representations from intermediate layers of NLMs (§9.1), irregularities across models and variants (§9.2), and potential downstream applications of our sense embeddings focusing on knowledge integration (§9.3).

Finally, in Section 10 we present our concluding remarks, and provide details about our release of sense embeddings, code and more.

2. Preliminaries

This work exploits the interaction between vector-based semantic representations (§2.1), recent developments on NLMs (§2.2), and curated sense inventories (§2.3). In this section we provide some background on these topics.

2.1. Vector Semantics

Nearly a century ago, Firth (1935) postulated that "the meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously". Indeed, after working on formal theories of word meaning definition, Wittgenstein (1953) conceded "the meaning of a word is its use in a language". This view of meaning representation became known as the Distributional Hypothesis (Harris, 1954, DH), which proposes that words that occur in the same contexts tend to have similar meanings. During this period, Osgood et al. (1957) further proposed representing the meaning of words as points in multi-dimensional space, with similar words having similar representations, thus being placed closely in this space. Still, it would take a few more decades of computing advancements to appreciate the implications of the DH.

Early VSMs. After some early works introducing vector space models (VSMs) for information retrieval (Salton 1971; 1975), Deerwester (1989; 1990) was the first to use dense vectors to represent word meaning, initially with a method called Latent Semantic Indexing (LSI), and later with Latent Semantic Analysis (LSA). LSA was based on a word-document weighted frequency matrix from which the first 300-dimensions resulting from Singular-Value Decomposition (SVD) would correspond to word embeddings. Lund & Burgess (1996) introduced another influential method similar to LSA, called Hyperspace Analogue to Language (HAL) which differed from LSA by considering word-word frequencies instead, introducing the notion of a fixed-sized window as context (e.g., the two words to the left and to the right) instead full documents, which would become the standard representation of context. Following these developments, Landauer & Dumais (1997) evaluated the performance of LSA embed-

dings learned from large corpora on a simple semantic task (synonymy tests) and found that these embeddings performed comparably to school-aged children, when measuring similarity between word pairs as the cosine similarity between their corresponding embeddings (inspired by applications for information retrieval). Already in this early period, Schutze (1992) and Yarowsky (1995) realized the potential for WSD applications based on the similarity between unsupervised word embeddings. Blei et al. (2003) would later introduce Latent Dirichlet Allocation (LDA) which uses a generative probabilistic approach to generalize and improve on the approach used for LSA, being widely adopted for topic modelling and other applications beyond semantic analysis.

Neural Models. Having established that corpus-based word embeddings are able to capture semantic knowledge, additional progress followed swiftly. A milestone in the evolution of word embeddings was the discovery that Neural Language Models (NLMs) implicitly develop word embeddings when training for the task of word prediction (Bengio et al., 2003b). Shortly after, Collobert 2007, 2008, 2011 demonstrated that word embeddings could be incorporated into neural architectures for various NLP tasks. With word2vec, Mikolov et al. (2013b) distilled the components of NLMs responsible for learning word embeddings into a lightweight and scalable solution, allowing this neural-based solution to be employed on corpora of unprecedented size (100B tokens). Nevertheless, count-based solutions would still remain important, particularly GloVe (Pennington et al., 2014), as these methods were also significantly improved. The next major improvement was the introduction of fastText (Bojanowski et al., 2017), which was able to represent words absent from training data by leveraging subword information, as well as refining several aspects of word2vec's training method.

Sense Embeddings. In spite of their success, word2vec, GloVe and fastText conflated different senses of the same word form into the same representation, a shortcoming known as the Meaning Conflation Deficiency (Camacho-Collados & Pilehvar, 2018). While a number of extensions were proposed for the creation of sense-specific representations, such as AutoExtend (Rothe & Schütze, 2015),

NASARI (Camacho-Collados et al., 2016), DeConf (Pilehvar & Collier, 2016) or Probabilistic FastText (Athiwaratkun et al., 2018), this issue would require the development of a new generation of NLMs in order to be effectively addressed.

2.2. Neural Language Modelling

The first major step towards contextual embeddings from NLMs, was the development of context2vec (Melamud et al., 2016), a single-layer bidirectional LSTM trained with the objective of maximizing similarity between hidden states and target word embeddings, similarly to word2vec. Peters et al. (2018a) built upon context2vec with ELMo, a deeper bidirectional LSTM trained with language modelling objectives that produce more transferrable representations. Both context2vec and ELMo emphasized WSD applications, providing the most convincing accounts until then that sense embeddings can be effectively represented as centroids of contextual embeddings, showing 1-NN solutions to WSD tasks that rivalled the performance of task-specific models.

With the introduction of highly-scalable Transformer architectures (Vaswani et al., 2017), two kinds of very deep NLMs emerged: causal (or left-to-right) models, epitomized by the Generative Pre-trained Transformer (Brown et al., 2020, GPT-3), where the objective is to predict the next word given a past sequence of words; and masked models, where the objective is to predict a masked (i.e., hidden) word given its surrounding words, of which the most prominent example is the Bidirectional Encoder Representations from Transformers (Devlin et al., 2019, BERT). The difference in training objectives results in these two varieties of NLMs specializing at different tasks, with causal models excelling at language generation and masked models at language understanding.[1]

BERT proved highly successfully at most NLP tasks (Rogers et al., 2020), and motivated the development of numerous derivative models, many of which we also explore in this work. In spite of this progress, Transformer-based NLMs can still show strong reliance on surface features (McCoy et al., 2019) and social

[1] Although recent models like BART (Lewis et al., 2020) show progress towards both.

biases which are hard to correct (Zhou et al., 2021). There are known theoretical limits to how much language understanding can be expected from models trained with language modelling objectives alone (Bender & Koller, 2020; Merrill et al., 2021), and it is not clear how far current models are from those limits.

2.3. Sense Inventories

The currently most popular English word sense inventory is the Princeton WordNet (Fellbaum, 1998) (henceforth, WordNet), a large semantic network comprised of general domain concepts curated by experts[2].

The core unit of WordNet is the synset, which represents a cognitive concept. Each lemma (word or multi-word expression) in WordNet belongs to one or more synsets, and word senses amount to the combination of word forms and synsets (referred as sensekeys). As a result, the set of words that belong to a synset can be described as synonyms, with some words being ambiguous (belonging to additional synsets) while others not (specific to a synset). The predominant semantic relation in WordNet, which relates synset pairs, is hypernymy (i.e., Is-A). Each synset also features a gloss (dictionary definition), part-of-speech (noun, verb, adjective or adverb) and lexname[3], which is a syntactic category and logical grouping. Synsets are formally represented as numerical codes. Following related works, we also represent them using the more readable format $lemma^{\#}_{POS}$, where *lemma* corresponds to synset's most representative lemma.

As an example, the lemma 'mouse' is polysemous belonging to the $mouse^{1}_{n}$ (rodent) and $mouse^{4}_{n}$ (computer mouse) synsets, among others. The most frequent sense for mouse, mouse%1:05:00:: (sensekey), belongs to the synset $mouse^{1}_{n}$ (02330245n) which has an hypernymy relation with $rodent^{1}_{n}$, lexname 'noun.animal', and gloss "any of numerous small rodents typically [...]".

Following Loureiro & Jorge (2019a), we use WordNet version 3.0, which contains 117,659 synsets, 206,949 senses, 147,306 lemmas, and 45 lexnames.

[2]Babelnet (Navigli & Ponzetto, 2010), Wiktionary (Meyer & Gurevych, 2012) and HowNet (Dong et al., 2006) are popular alternatives covering other languages.

[3]Lexnames are also known as supersenses (Flekova & Gurevych, 2016; Pilehvar et al., 2017).

3. Related Work

In this section we cover related work on the various well-researched topics that our work intersects, namely Sense Embeddings (§3.1), WSD (§3.2) and Probing NLMs (§3.3).

3.1. Sense Embeddings

Sense embeddings emerged in NLP due to the so-called meaning conflation deficiency of word embeddings (Camacho-Collados & Pilehvar, 2018). By merging several meanings into a single representation, the single vector proved insufficient in certain settings (Yaghoobzadeh & Schütze, 2016), and contradicted common laws in distance metrics, such as the triangle inequality (Neelakantan et al., 2014). In order to solve this issue, the field of sense vector representation mainly split into two categories: (1) unsupervised, where senses were learned directly from text corpora (Reisinger & Mooney, 2010; Huang et al., 2012; Vu & Parker, 2016); (2) or knowledge-based, where senses were linked to a predefined sense inventory by exploiting an underlying knowledge resource (Rothe & Schütze, 2015; Pilehvar & Collier, 2016; Mancini et al., 2017; Colla et al., 2020a).

In this article, we focus on the latter type of representation, particularly leveraging powerful Transformer-based language models trained on unlabeled text corpora. As such, the final representation is mainly constructed based on the knowledge learned by the language models, and knowledge resources such as WordNet serve to guide the annotation process. The goal of this paper is indeed to construct a task-agnostic sense representation that can be leveraged in semantic and textual applications. This differs from traditional static sense embeddings which, with a few notable exceptions (Li & Jurafsky, 2015; Flekova & Gurevych, 2016; Pilehvar et al., 2017), were mainly leveraged in intrinsic sense-based tasks only. As we show throughout this paper, general-purpose sense representations learned with the power of Transformers and guided through an underlying lexical resource such as WordNet prove to be robust in a range of text-based semantic tasks, as well as in intrinsic sense-based benchmarks.

3.2. Word Sense Disambiguation

As one of the earliest Artificial Intelligence tasks, WSD has a long history of research. In this work, our coverage of related work for WSD is focused on recent systems using Transformer-based architectures for two reasons: our own experiments are also focused on Transformer-based systems; the current state-of-the-art for WSD has converged on these systems. Additionally, we also distinguish between solutions addressing WSD from the nearest neighbors paradigm, using pre-computed sense embeddings, and task-specific solutions fine-tuning Transformer models or training classifiers using their internal representations.

3.2.1. Nearest Neighbors

Our prior LMMS work (described throughout this paper) was the first to demonstrate that a nearest neighbors solution based on sense embeddings pooled from internal representations of BERT (i.e., feature extraction) could clearly outperform the state-of-the-art of the time, which still had not adopted Transformer-based models.

SensEmBERT (Scarlini et al., 2020a) followed a similar approach to LMMS, but leveraged BabelNet to reduce dependency on annotated corpora, producing sense embeddings that performed better on WSD, though limited to nouns only.

With ARES, Scarlini et al. (2020b) introduce a method to produce a large number of semi-supervised annotations to dramatically increase the coverage of the sense inventory, and demonstrated that sense embeddings learned from those annotations can perform substantially better on WSD than LMMS.

SensEmBERT and ARES use the same layer pooling method and gloss embeddings as LMMS, although both have employed not only BERT-L, but also its multilingual variant, showing strong performance on languages other than English as well.

In addition to WSD, to our knowledge, the only other task these works have applied their sense embeddings is Word-in-Context (Pilehvar & Camacho-Collados, 2019, WiC), which we also address in this work.

3.2.2. Trained Classifiers

When it comes to using Transformers to train classifiers specific to the WSD task, we encounter a much more diverse set of solutions in comparison to feature extraction approaches.

One of the earliest and most straightforward supervised classifiers for WSD using BERT was the Sense Vocabulary Compression (SVC) of Vial et al. (2019), which added layers to BERT, topped with a softmax classifier, to be trained targeting a strategically reduced set of admissible candidate senses.

Following outstanding results on a range of text classification tasks by model fine-tuning, GlossBERT (Huang et al., 2019) fine-tuned BERT using glosses so that WSD could be framed as a text classification task pairing glosses to words in context. KnowBERT (Peters et al., 2019) employs a more sophisticated fine-tuning approach, designed to exploit knowledge bases (WordNet and Wikipedia) as well as glosses.

Straying further from prototypical classifiers, Blevins & Zettlemoyer (2020) (BEM) propose a bi-encoder method which learns to represent senses based on glosses while performing the optimization jointly with the underlying BERT model. Taking advantage of an ensemble of sense embeddings from LMMS and SensEmBERT, along with additional resources, EWISER (Bevilacqua & Navigli, 2020) trains a multifaceted high performance WSD classifier.

Finally, the current state-of-the-art for WSD is ConSeC (Barba et al., 2021), which obtains impressively strong results by framing WSD as an extractive task, similar to extractive question answering, trained through fine-tuning BART (Lewis et al., 2020), a sequence-to-sequence Transformer which outperforms BERT on reading comprehension tasks (while being of comparable size).

In Loureiro et al. (2021) we extensively compared fine-tuning and feature extraction approaches for the WSD task. Consistent with prior work, we found that fine-tuning overall outperforms feature extraction. However, under comparable circumstances, the performance gap is narrow and feature extraction shows improved few-shot performance and less frequency bias.

3.3. Probing Neural Language Models

As NLMs became popular, investigating properties of their internal states, or intermediate representations, also became an important line of research, often referred to as 'model probing'. Probing operates under the assumption that if a relatively simple classifier, based exclusively on representations from NLMs, can perform well at some task, then the required information was already encoded in the representations. For clarity, we define probes as functions (learned or heuristic) designed to reveal some intrinsic property of NLMs. In this section we cover probing works focused on lexical semantics and layer-specific variation that inspired our probing analysis. We distinguish these works by their use of probes trained using representations (learned), and probes directly comparing or analysing unaltered representations (heuristics, such as nearest neighbors).

3.3.1. Learned Probes

Among the most influential findings in this line of research was the discovery by Hewitt & Manning (2019) that syntactically valid parse trees could be uncovered from linear transformations of word representations obtained from pre-trained ELMo and BERT models. Motivated by this discovery, Reif et al. (2019) performed additional experiments focused on sense representation, including showing that a nearest neighbors based on BERT representations could outperform the reported WSD state-of-the-art, particularly when following Hewitt & Manning (2019)'s methodology to learn a probe tailored to sense representation. To increase sensitivity to sense-specific information, Reif et al. (2019) used a loss that considered the difference between the average cosine similarity of embeddings of words with the same senses, and embeddings of words with different senses. Both Hewitt & Manning (2019) and Reif et al. (2019) approaches are designed for probing representations obtained from single layers.

With ELMo, Peters et al. (2018a) introduced contextualized word representations that are obtained from a linear combination of representations from all layers of the model. This linear combination uses task-specific weights learned through an optimization process, often referred to in the literature as "scalar

mixing", and produced better results in downstream tasks when compared to representations obtained from individual layers. On closer inspection, Peters et al. (2018b) concluded that top layers can be less effective for semantic tasks possibly due to specialization for the language modelling tasks optimized during pre-training.

Tenney et al. (2019b) proposed an "edge probing" methodology, using scalar mixing, that allowed for evaluating different syntactic or semantic properties using a common classifier architecture, where probing models are trained to predict graph edges independently. In Tenney et al. (2019a), edge probing was employed to reveal that BERT implicitly performed different steps of a traditional NLP pipeline, in the expected order as information flows through the model, with lower layers processing local syntax (e.g., Part-of-Speech) and higher layers processing complex semantics of arbitrary distance (e.g., Semantic Roles). Raising concerns about remaining faithful to the information encoded in the representations, Kuznetsov & Gurevych (2020) proposes reducing the expressive power of learned probes while improving edge probing.

Liu et al. (2019a) ran several probing experiments with simpler probes (i.e., linear classifiers), investigating differences between NLM architectures, namely ELMo, GPT and BERT, while still finding competitive performance with state-of-the-art task-specific models. They confirm that LSTM-based models (i.e., ELMo) present more task-specific (less transferable) top layers, but Transformers-based models (i.e., BERT) are less predictable and do not exhibit monotonic increase in task-specificity, in line with our own findings. GPT was found to significantly underperform ELMo and BERT, which Liu et al. (2019a) attributes to the fact that GPT is trained unidirectionally (left-to-right), while ELMo and BERT are trained bidirectionally.

3.3.2. Representational Similarity

Without recourse to learned probes, Ethayarajh (2019) investigated differences between ELMo, GPT-2 and BERT, relying on experiments based on cosine similarity to learn about the context-specificity of their representations. Etha-

yarajh (2019) found that top layers show highest degree of context-specificity, but all layers of all three models produced highly anisotropic representations, with directions in vector space confined to a narrow cone, concluding that this property is an inherent consequence of the contextualization process. The anisotropy observed for all contextualized NLMs also supports the hypothesis of Reif et al. (2019) that sense-level information is encoded in a low-dimensional subspace, since contextualization is crucial for sense disambiguation.

Vulić et al. (2020) reached similar conclusions regarding the detrimental contribution of top layers for lexical tasks (e.g., lexical semantic similarity) while also finding improved results from averaging different layers, particularly task-specific layer subsets, prompting further research into layer weighting or meta-embedding approaches, and motivating the present work. Through direct comparison of cosine similarities, Chronis & Erk (2020) reached similar conclusions as Vulić et al. (2020) about the role of top layers for lexical similarity tasks, adding that top layers appear to better approximate relatedness than similarity.

Voita et al. (2019a) probed Transformer-based NLMs from an Information-Bottleneck perspective to learn about differences in information flow across the network according to language modelling pre-training objectives, particularly left-to-right, MLM, and translation. They find that the MLM objective induces representation of token identity in the lower layers, followed by a more generalized token representation in intermediate layers, and then token identity information gets recreated at top layers.

Mickus et al. (2020) specifically verified whether BERT representations comprise a coherent semantic space. These experiments are explicitly detached from learned probes, as Mickus et al. (2020) explains that such methodology interferes with direct assessment of the coherence of the semantic space as produced by NLMs. Using cluster analyses, they find that BERT indeed appears to represent a coherent semantic space (based only on representations from the final layer), although its Next Sentence Prediction (NSP) modelling objective leads to encoding semantically irrelevant information (sentence position), corrupting similarity relationships and complicating comparisons with other NLMs.

4. Method

We propose a principled approach for sense representation based on contextual NLMs trained exclusively with self-supervision. This approach is an extension of Loureiro & Jorge (2019a), addressing relevant issues still largely unresolved, particularly the influence of embeddings from the different layers composing NLMs, with the introduction of a novel layer probing methodology. Moreover, in this work, we reinforce the distinction between sense disambiguation and sense matching by introducing methodological differences specific to each application scenario.

This section starts by explaining the methods used in Loureiro & Jorge (2019a) for learning (§4.1), extending (§4.2) and applying sense embeddings (§4.3). Afterwards, we introduce our proposed layer probing methodology (§4.4), including how the resulting analysis informs a grounded pooling operation for combining embeddings from all layers of a NLM.

4.1. Learning Sense Embeddings

The initial process to learn sense embeddings is based on sense-annotated sentences and contextualized embeddings of annotated words or phrases in context. An overview of the process can be seen at Figure 1.

Formally, in order to generate sense embeddings learned in context from natural language, we require a pre-trained contextual NLM Ω (frozen parameters) and a corpus of sense-annotated sentences S. Every sense ψ is represented from the set of contextual embeddings $\vec{c_l} \in C_\psi$, obtained by employing Ω on the set of sentences S_ψ annotated with that sense (considering only contextual embeddings specific to tokens annotated with sense ψ), using representations at each layer $l \in L$, such that:

$$\vec{\psi} = \frac{1}{|C_\psi|} \sum_{l \in L} \sum_{\vec{c} \in C_\psi} \vec{c_l} \ , \ \textit{where } C_\psi = \Omega(S_\psi) \tag{1}$$

The L set of layers typically used for sense representation is the last four $[-1, -2, -3, -4]$ (reversed layer indices), as discussed in Section 3.1.

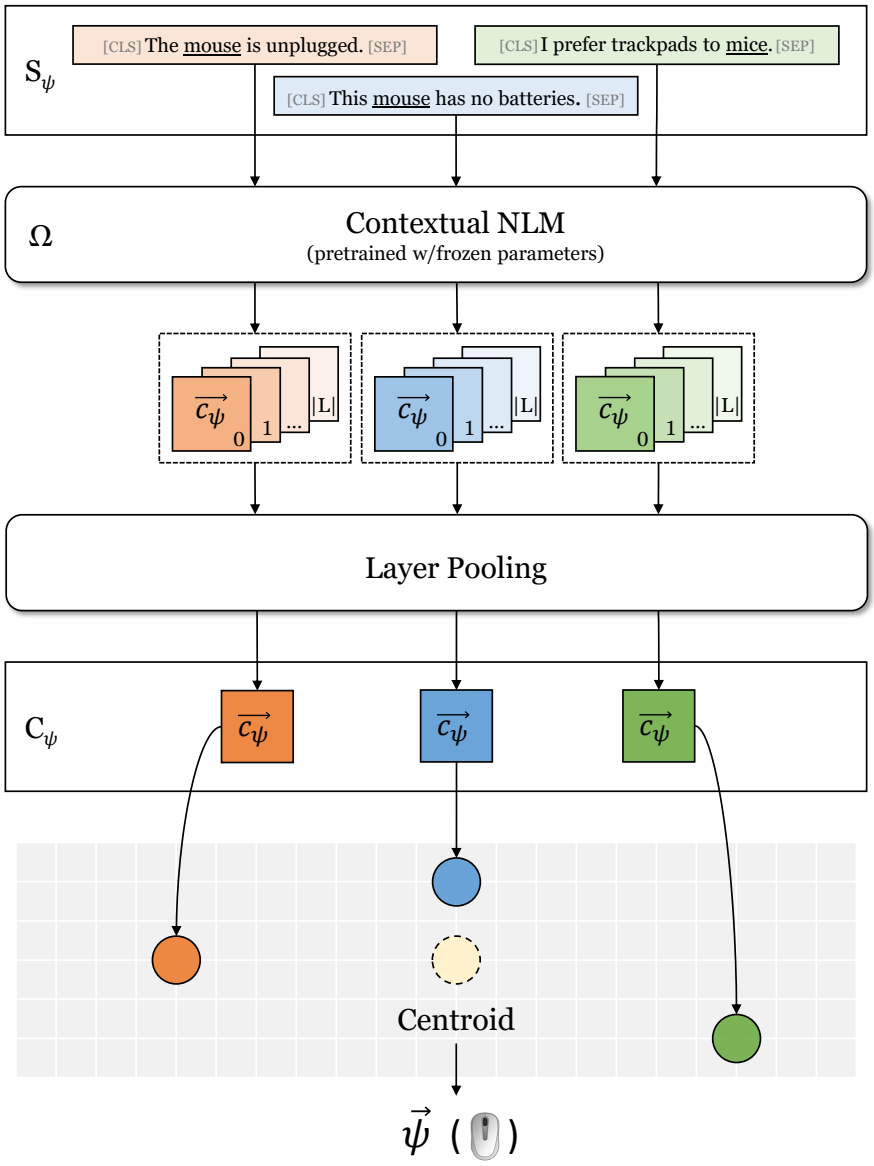


Figure 1: Overview of learning sense embeddings from annotated corpora. Showing how the sense ψ for 'computer mouse' is determined from a set for sentences annotated with that sense S_ψ (padded with special tokens as expected by the NLM Ω). After pooling contextual embeddings C_ψ from layers L, the sense embedding for $\vec{\psi}$ is computed as the centroid of C_ψ.

Contextual NLMs typically operate at the subword-level, so the token-level embeddings $\vec{c}$ produced by Ω correspond to the average of each token's subword contextual embeddings (depending on the NLM, these may be BPE or WordPiece embeddings). Similarly, whenever sense-annotations cover a span of several tokens, we also use the average of the corresponding token-level embeddings as the contextual embedding. Contextual NLMs are pre-trained using special tokens at specific locations, so we also include these tokens in their expected positions (e.g., `[CLS]` at the start and `[SEP]` at the end with BERT).

As described in Section 2.3, WordNet can be used to represent senses in two ways: sensekeys and synsets. Sense-annotated corpora most often use sensekey annotations, so in those cases sensekey embeddings do not require any intermediate mapping. Synset embeddings can be derived from sensekey annotations in at least two ways, which we differentiate as 'direct' and 'indirect'. In the direct approach, each sensekey annotation is converted (mapped) to the corresponding synset, so synset representations are learned from each annotation instance. In the indirect approach we first learn sensekey-level embeddings, without converting annotations, and afterwards compute synset embeddings as the average of corresponding sensekey embeddings. The latter approach has been explored in earlier works in sense embeddings (Rothe & Schütze, 2015). In this work we explore both approaches.

4.2. Extending Coverage with Additional Resources

Given that one of the major issues in supervised WSD is the lack of sense annotations (Pasini, 2020), not just in their quantity but also in terms of their coverage of the sense inventory, we require solutions to address this in our method. On this matter, we also follow the methods we first proposed in Loureiro & Jorge (2019a) and later optimized with the introduction of the UWA corpus in Loureiro & Camacho-Collados (2020). The two methods, ontological propagation and gloss representation, are designed to reach full coverage of the sense inventory, and they are complementary by exploiting different resources, namely semantic relations between senses and glosses (combined with lemmas).

4.2.1. Ontological Propagation

In Section 2.3 we introduced WordNet and the different elements and relations composing this semantic network. The ontological propagation method we presented in Loureiro & Jorge (2019a) exploits these relations between senses in WordNet in order to infer embeddings for senses which may not occur in annotated corpora. It is possible to infer accurate sense embeddings from these relations due to the fine-granularity of WordNet, along with widespread synonymy and hypernymy relations, to the extent that in Loureiro & Camacho-Collados (2020) we showed that even annotations for unambiguous words can significantly improve the propagation process.

Since available corpora do not provide full-coverage annotations for our sense inventory of interest, by following the process described in Section 4.1 we are left with a represented senses Ψ, and a set of unrepresented senses Ψ'. The propagation process involves three steps, using increasingly abstract relations from WordNet - sets of synonyms (synsets), hypernymy relations, and lexical categories (lexnames or supersenses). In case we are targeting synset-level representations, then the first step/level should be skipped.

Considering we are provided mappings between sensekeys, synsets, hypernyms and lexnames, we infer Ψ' iteratively following Algorithm 1. After each of these sequential steps, every inferred $\vec{\psi}$ is added to the set of represented senses Ψ. This propagation method ensures full-coverage provided that initial sense embeddings Ψ are sufficiently diverse such that falling back on propagating from lexnames (supersenses) is always possible.

Since this method is designed to achieve full-representation of the sense inventory based on a subset of senses observed in context, the inferred representations are also of a similar contextual nature. However, unless the initial set of sense embeddings is nearly complete, and particularly diversified, this propagation method produces some number of identical representations for distinct senses, which is most undesirable for disambiguation applications, and to a lesser extent, sense matching applications as well.

Algorithm 1: Propagation method to infer unrepresented senses Ψ', using sense embeddings Ψ learned from annotations, and relations R.

Propagate (Ψ, Ψ', R)

  foreach *unrepresented sense* $\psi' \in \Psi'$ **do**

    $R_{\psi'} \leftarrow \{\text{all represented } \vec{\psi} \in \Psi \text{ for which } (\psi, \psi') \in R\}$;

    if $|R_{\psi'}| > 0$ **then**

      $\vec{\psi'} \leftarrow$ average of sense embeddings in $R_{\psi'}$;

      `Insert(`$\vec{\psi'}, \Psi$`);` `// add to represented`

      `Remove(`ψ', Ψ'`);` `// remove from unrepresented`

  return Ψ, Ψ';

$\Psi, \Psi' \leftarrow$ **Propagate** $(\Psi, \Psi', \{all\ (\psi, \psi') : \texttt{Synset}(\psi) = \texttt{Synset}(\psi')\})$

$\Psi, \Psi' \leftarrow$ **Propagate** $(\Psi, \Psi', \{all\ (\psi, \psi') : \texttt{Hypernym}(\psi) = \texttt{Hyper.}(\psi')\})$

$\Psi, \Psi' \leftarrow$ **Propagate** $(\Psi, \Psi', \{all\ (\psi, \psi') : \texttt{Lexname}(\psi) = \texttt{Lexname}(\psi')\})$

4.2.2. Leveraging Glosses and Lemmas

In Loureiro & Jorge (2019a) we introduced a method for representing sense embeddings based on glosses and lemmas. This method is inspired by a typical baseline approach used in works pertaining to sentence embeddings, and it amounts to simply averaging the contextual embeddings for all tokens present in a sentence. In our case, we use glosses as sentences, but also introduce lemmas into the gloss' context. By combining glosses with lemmas, we not only augment the information available to represent senses, but we are also able to generate sense embeddings which are lemma-specific (sensekey-level), instead of only concept-specific (synset-level) if we only used glosses. As such, sense embeddings generated by this method address the redundancy issue arising from the previously described propagation method, while simultaneously introducing representational information which is complementary to contextual embeddings extracted from sense-annotated sentences.

The method proceeds as follows. For every lemma/sense pair (i.e., sensekey)

in a sense inventory, we build the template "<*lemma*> , <*sense lemmas*> - <*sense gloss*>". For instance, based on WordNet, the synset race_2^v has the lemmas *race* and *run* which are provided with the following sensekey-specific fill-outs of the template:

- *race%2:33:00::* - "race - run, race - compete in a race"
- *run%2:33:01::* - "run - run, race - compete in a race"

The initial "<*lemma*>" component of the template can be omitted if the target representation level is synsets, as it only serves the purpose of reinforcing the lemma which is specific to the sensekey. The templated string is processed by Ω, similarly to sentences S in Section 4.1, but here we use the resulting set of contextual embeddings for every token $C_\star$.

Considering that we have a complete set of sense embedding Ψ, based on sense annotations and propagation as described in Sections 4.1 and 4.2.1, we augment $\forall\vec{\psi} \in \Psi$ with gloss and lemma information as follows:

$$\vec{\psi} = \frac{1}{2}(||\vec{\psi}||_2 + ||\frac{1}{|C_\star|}\sum_{l \in L}\sum_{\vec{c} \in C_\star}\vec{c}_l||_2) \text{ , where } C_\star = \Omega(\text{Template}(\psi)) \quad (3)$$

In contrast to Loureiro & Jorge (2019a), which proposed using concatenation to merge this new set of sense embeddings based on glosses and lemmas with the previously mentioned set, in this work we propose merging through averaging instead. This departure is motivated by the fact that Loureiro & Jorge (2019a) found that while concatenation outperformed averaging for WSD, the difference in performance was modest, and in this work we are interested in additional tasks which that work did not cover. Merging representations through concatenation doubles the dimensionality of sense embeddings, increasing computational requirements and complicating comparison with contextual embeddings, among other potential applications. On the other hand, merging representations through averaging allows for adding more components while retaining a similar vector, of equal dimensionality to contextual embeddings, and represented in the same vector space.

4.3. Applying Sense Embeddings

In this section we address how sense embeddings can be employed for solving various tasks, grouped under two paradigms: disambiguation and matching. Disambiguation assigns a word in context (i.e., in a sentence) to a particular sense out of a subset of candidate senses, restricted by the word's lemma and part-of-speech. Matching also assigns specific senses to words, but imposes no restrictions, admitting every entry in the sense inventory for each assignment.

The different conditions for disambiguation and matching require sense representations with different degrees of lexical information and semantic coherence. Whereas, for disambiguation, lexical information can be absent from sense representations, due to the subset restrictions, for matching, lexical information is essential to distinguish between word forms carrying identical or similar semantics. Similarly, the disambiguation setting has no issues with sense representations displaying inconsistencies such as *eat* being more similar to *sleep* than to *drink*, since these all belong to disjoint subsets, but the order and coherence of these similarities is relevant for sense matching applications. This distinction leads us to specialize sense embeddings accordingly in Section 4.4.

To disambiguate a word w in context, we start by creating a set Ψ_w of candidate senses based on its lemma and part-of-speech, using information provided with the sense inventory. Afterwards, we compute the cosine similarities (denoted 'cos') between the word's contextual embedding $\vec{c}_w$ and the pre-computed embeddings for each sense in this subset Ψ_w (both using the same layer pooling). Finally, we assign the sense whose similarity is highest (i.e., nearest neighbor):

$$\Psi_w = \{\vec{\psi} \in \Psi : \text{lemma and part-of-speech of } \psi \text{ match } w\}$$
$$\text{Disambiguation}(w) = \arg\max_{\vec{\psi} \in \Psi_w} (\cos(\vec{c}_w, \vec{\psi})) \tag{4}$$

To match a word w in context, without restrictions, we follow the approach for disambiguation but simply consider the full sense inventory Ψ instead of Ψ_w:

$$\text{Matching}(w) = \arg\max_{\vec{\psi} \in \Psi} (\cos(\vec{c}_w, \vec{\psi})) \tag{5}$$

4.4. Grounding Layer Pooling

Up until this point, we have described our method closely following our prior work in Loureiro & Jorge (2019a). As we covered in earlier sections, NLMs can show substantial and, more importantly, unexpected variation in task performance across their layers. Considering this work's focus on a more principled and grounded focus on sense representation with NLMs, our methodology also covers this important aspect.

In this section we present two methods targeting the layers composing NLMs. The first method probes each layer's adeptness for sense representation. Consequently, the second method in this section is designed to capitalize on that knowledge towards sense representations which better capture NLM's ability to represent senses over the current paradigm.

As we alluded to in Section 4.3, sense representation should be viewed in light of the intended applications for these representations. In particular, in this work we differentiate between representations used for disambiguating words, and for matching or comparing senses. This distinction is motivated by the fact that disambiguation, which is the prevalent sense-related task on NLP, only requires that sense representations be adequately differentiated between the restricted set of senses which share the same lemmas and parts-of-speech. However, there exist other potential applications where sense representations are matched without any constraints on the sense inventory, and thus require that senses be coherently represented across the semantic space.

4.4.1. Sense Probing

In order to assess the contribution of individual layers of a pre-trained NLM for sense representation, we directly evaluate the performance of representations from these layers on tasks related to the previously described disambiguation or matching scenarios. These tasks are solved using the nearest neighbors approaches described in Section 4.3, comparing pre-computed sense embeddings with contextual embeddings obtained from the same layer.

For this probing experiment, we follow the method for learning sense representations described in Section 4.1, but create multiple sets of senses Ψ_l for each layer l in the NLM. To maintain focus on assessing the performance of representations learned directly from specific layers, we ensure that test instances have all their senses represented in the sense-annotated corpora used to precompute Ψ_l, $\forall l \in L$. Thus, our probing experiments do not use techniques to infer or enrich sense representations, such as those we described in Section 4.3, which could otherwise act as confounders.

The resulting performance scores for every layer $l \in L$ composing a specific Ω, using a corresponding Ψ_l, not only reveal which layers perform best, but also inform the layer pooling method described next.

4.4.2. Sense Profiling

We use the probing results described earlier as the basis for a pooling operation which is better grounded than the current paradigm of using the sum of the last four layers, and also better performing as we show later in this work. We designate each set of model-specific layer weights as a 'sense profile', and consider distinct sense profiles for disambiguation and matching, depending on the choice of disambiguation or matching tasks during layer probing.

These proposed sense profiles are a more immediate version of the Scalar Mixing used in Tenney et al. (2019a), being based on heuristically-derived sets of layer weights, instead of learning them through task optimization. Considering this, we understand sense profiles to be closer to the extraction configurations of Vulić et al. (2020).

Granted we have performance scores s_l $\forall l \in L$, for a specific Ω, we obtain layer specific weights w_l $\forall l \in L$ by applying the softmax function:

$$w_l = \frac{\exp(s_l/t)}{\sum_{l' \in L} \exp(s_{l'}/t)} \tag{4}$$

We use the temperature scaling parameter t to skew the weight distribution towards highest performing layers. While simple, temperature scaling has been found surprisingly effective at calibrating neural network predictions (Guo et al.,

2017). This parameter is to be determined empirically and is only specific to application settings, not models.

In Table 1 we demonstrate the interaction between performance scores and layer weights conditioned on the temperature parameter. In that table, and others found in this work, we use reverse layer indices so that we can consistently refer to the final layer of any model using the -1 index, regardless of the number of layers in the NLM.

Consequently, we employ sense profiles comprised of weights $w_l \; \forall l \in L$ to retrieve contextual embeddings from Ω, and generate our sense embeddings accordingly, updating formula (1) such that:

$$\vec{\psi} = \frac{1}{|C_\psi|} \sum_{l \in L} \sum_{\vec{c} \in C_\psi} w_l * \vec{c}_l \;\; , \; where \; C_\psi = \Omega(S_\psi) \tag{5}$$

This set of sense embeddings learned from annotations using sense profiles, undergoes the same extensions and augmentations described earlier (§4.2).

To be clear, the process of probing layer performance and determining sense profiles to pool contextual embeddings from all layers (including when learning sense embeddings from annotations) is carried out for both the disambiguation and matching settings independently. As a result, we produce two sets of sense embeddings for each NLM based on sense profiles, which we distinguish from the LMMS sense embeddings introduced in Loureiro & Jorge (2019a) as LMMS-SP (Sense Profiles). The SP-WSD (for Word Sense Disambiguation) and SP-USM (for Uninformed Sense Matching) abbreviations are used to refer to sense embeddings based on disambiguation and matching sense profiles respectively.

	INIT	-24	-23	-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
BERT-L	53	58	62	63	65	67	68	68	69	70	71	71	71	71	72	72	71	72	73	72	73	74	**75**	**75**	72
XLNet-L	51	57	65	67	68	70	71	72	**73**	**73**	72	72	72	72	71	71	71	71	71	71	72	72	72	72	68
RoBERTa-L	53	57	63	66	67	69	71	72	73	73	74	74	74	74	**75**	**75**	**75**	74	**75**	74	74	74	73	74	71
ALBERT-XL	54	65	67	68	69	70	70	**71**	**71**	**71**	**71**	**71**	**71**	**71**	**71**	70	70	69	69	69	69	69	69	69	64

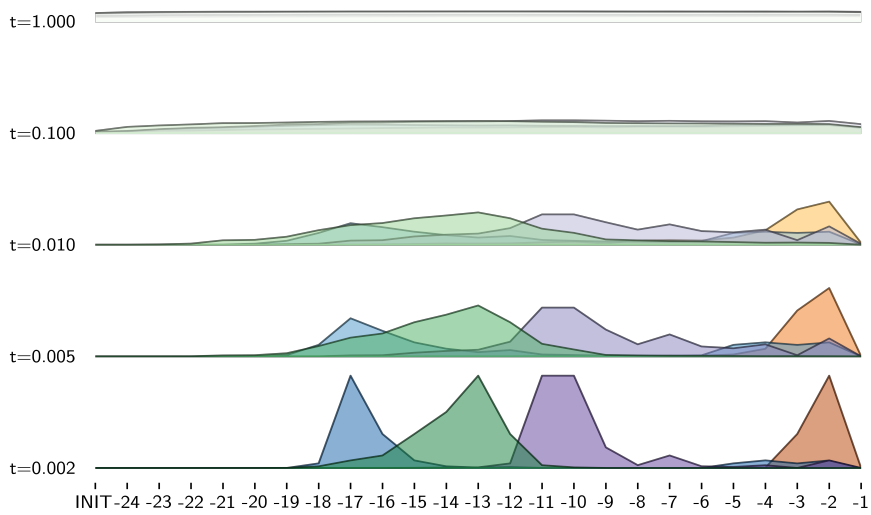


Table 1: Shows interaction between F1 scores (rounded) for 1NN WSD using each layer of four different NLMs, and respective weight distributions (matching colors) using decreasing temperature parameters. Lower temperatures induce higher skewness towards layers that perform best on the probing validation set. Distributions based on t=1.000 are almost uniform, while t<0.002 would place almost all mass on single best layer.

5. Experimental Setting

In this section we provide details about our experimental setting, including a description of the models (§5.1) and datasets (§5.2) used for learning sense representations.

5.1. Transformer-based Language Models

In this work we experiment with several Transformer-based Language Models, including all the original English BERT models released by Devlin et al. (2019) as well as several other BERT-inspired alternatives, namely XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019b) and ALBERT (Lan et al., 2020). This section briefly describes the most relevant features of each of these models for our use case. We summarize the differences between each variant of these models on Table 2.

BERT. The model released by Devlin et al. (2019) is first prominent Transformer-based NLM designed for language understanding. It is pre-trained with two unsupervised modelling objectives, Masked Language Modelling (MLM) and Next Sentence Prediction (NSP), using English Wikipedia and BookCorpus (Zhu et al., 2015). It uses WordPiece tokenization, splitting words into different components at the character-level (i.e., subwords). BERT is available in several models differing not only on parameter size, but also tokenization and casing. The 'whole-word' models were released after publication, showing slightly improved benchmark performance when trained with whole words being masked instead of subwords resulting from WordPiece tokenization.

XLNet. Based on a Transfomer-XL (Dai et al., 2019) architecture, Yang et al. (2019) release XLNet featuring Permutation Language Modelling (PLM) as the only pre-training objective. The motivation for PLM is that it does not rely on masked tokens, and thus makes pre-training closer to fine-tuning for downstream tasks. It is also trained on much larger corpora than BERT, adding a large volume of web text from various sources to the corpora used for BERT.

Instead of using WordPiece for tokenization, XLNet uses SentencePiece (Kudo & Richardson, 2018), which is a very similar open-source version of WordPiece.

RoBERTa. The model proposed by Liu et al. (2019b) is explicitly designed as an optimized version of BERT. RoBERTa does not use the NSP pre-training objective after finding that it deteriorates performance in the reported experimental setting, performing only MLM during pre-training. It is also trained with some different choices of hyperparameters (e.g., larger batch sizes) that improve performance on downstream tasks. The models released with RoBERTa are also trained on larger corpora composed mostly of web text, similarly to XLNet. As for tokenization, RoBERTa opts for byte-level BPE, following Radford et al. (2019), which makes retrieving embeddings for specific tokens more challenging (i.e., spacing must be explicitly encoded).

ALBERT. Aiming for a lighter architecture, Lan et al. (2020) propose ALBERT as a more parameter-efficient version of BERT. In spite of changes introduced to improve efficiency (e.g., cross-layer parameter sharing), ALBERT is based on a similar architecture to BERT. Besides improving efficiency, ALBERT also improves performance on downstream tasks by replacing NSP with the more challenging Sentence Order Prediction (SOP) objective. ALBERT uses the same SentencePiece tokenization as XLNet, and it is trained on similar corpora. It is released in several configurations, showing benchmark performance comparable to BERT while using fewer parameters.

The full set of 14 model variants detailed on Table 2 are only used for layer-specific validation performance on WSD and USM tasks. For task evaluation and analyses, we proceed with the single best performing model configuration from each model family, according to results from the validation experiments.

We use the Transformers package (Wolf et al., 2020) (v3.0.2) for experiments with BERT, XLNet and ALBERT, and the fairseq package (Ott et al., 2019) (v0.9.0) for experiments with RoBERTa [4].

[4]Initial experiments with RoBERTa showed slightly better results using fairseq.

Model	Configuration	Params.	Layers	Heads	Dims.	Tokenization	Tasks	Corpus
BERT	B	110M	12	12	768	WordPiece	MLM, NSP	16GB
	B-UNC	110M	12	12	768	WordPiece, Unc.	MLM, NSP	16GB
	L	340M	24	16	1024	WordPiece	MLM, NSP	16GB
	L-UNC	340M	24	16	1024	WordPiece, Unc.	MLM, NSP	16GB
	L-WHL	340M	24	16	1024	WordPiece	MLM, NSP	16GB
	L-UNC-WHL	340M	24	16	1024	WordPiece, Unc.	MLM, NSP	16GB
XLNet	B	110M	12	12	768	SentencePiece	PLM	158GB
	L	340M	24	16	1024	SentencePiece	PLM	158GB
RoBERTa	B	125M	12	12	768	Byte-level BPE	MLM	160GB
	L	355M	24	16	1024	Byte-level BPE	MLM	160GB
ALBERT	B	11M	12	12	768	SentencePiece	MLM, SOP	160GB
	L	17M	24	16	1024	SentencePiece	MLM, SOP	160GB
	XL	58M	24	16	2048	SentencePiece	MLM, SOP	160GB
	XXL	223M	12	64	4096	SentencePiece	MLM, SOP	160GB

Table 2: Feature comparison for the NLMs used in this work. Configuration names are shortened for readability: B - Base; L - Large; XL - Extra Large; XXL - Extra Extra Large; UNC - Uncased; WHL - Whole-Word.

5.2. Corpora for Training and Validation

We learn the initial set of sense representations described in Section 4.1 using sense-annotated corpora, namely SemCor (Miller et al., 1994) and the Unambiguous Word Annotations corpus (Loureiro & Camacho-Collados, 2020, UWA).

SemCor is a sense-annotated version of the Brown Corpus that still remains the largest corpus with manual sense-annotations despite its age. It includes 226,695 annotations for 33,362 sensekeys (25,942 synsets), reaching a coverage of 16.1% of WordNet's sense inventory. We use the version released in Raganato et al. (2017), which includes mappings updated to WordNet version 3.0.

UWA is our recently introduced corpus composed exclusively of annotations for unambiguous words from Wikipedia sentences. Since WordNet is mostly composed of unambiguous words, UWA not only allows for representing the majority of WordNet senses (56.7%, when combined with SemCor) from direct annotations, but also leads to improved sense representation for senses learned

through propagation (as described in Section 4.2.1), due to network effects. UWA is released in several versions of different sizes, in this work we use the version with up to 10 examples per sense (denoted UWA10), which includes 867,252 annotations for 98,494 sensekeys (67,860 synsets).

In order to avoid interference with the standard test sets, we perform our layer analysis and probing using a custom validation set, based on the MASC[5] corpus (Ide et al., 2010), following Loureiro et al. (2021). Considering that our layer experiments are focused on intrinsic properties of NLMs, this custom version of the MASC corpus is restricted to only include annotations for senses that occur in SemCor. Any sentence annotated with senses not occurring in SemCor is discarded, leaving a total of 14,645 annotations. As such, our layer experiments use sense embeddings learned from SemCor and validated using this restricted version of MASC, without requiring strategies for inferring senses (e.g., ontological propagation), or fallbacks (e.g., Most Frequent Sense).

6. Probing Analysis

In this section we present the outcome of the probing methodology described in Section 4.4 applied on the models detailed in Section 5.1. We report probing results in this section so they are presented and discussed before the evaluation and analysis sections, which report downstream task results using layer pooling informed by the probing analysis.

This section starts by covering our initial findings regarding layer performance variation patterns observed for all models (§6.1). Second, we present validation results using our proposed sense profiles for both disambiguation and matching scenarios (§6.2). Finally, we present our rationale for choosing which sense profile should be used according to the type of task (§6.3).

[5]We use the version of the MASC corpus released in Vial et al. (2018)

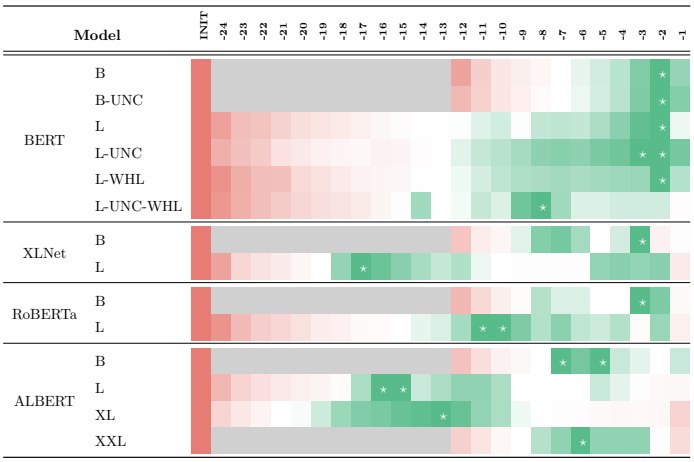
6.1. Variation in Layer Performance Across NLMs

As discussed in Section 3.3, it is well-understood that task performance varies considerably depending on which layers are used to retrieve embeddings from. While some works have analysed task performance per layer specifically for the task of WSD (Reif et al., 2019; Loureiro et al., 2021), there is still lacking an in-depth cross-model comparison.

In Table 3 we report WSD and USM performance for individual layers of each of the 14 models belonging to 4 different Transformer-based model families. These results are obtained using the methodology described in Section 4.4.1. We observe that indeed, the final layer (-1) is never optimal for either WSD or USM performance. More interestingly, we find that while the second-to-last layer (-2) always performs best for WSD in BERT models (with the exception of BERT-L-UNC-WHL), that pattern does not hold for the other models tested. For some models, such as XLNet-L or ALBERT-L, we even find that the best performing layers are closer to the initialization layer (INIT) than to the final layer. Another apparent pattern is that the best performing layers for USM are consistently lower than for WSD. This can be explained by the fact that USM benefits from lexical information encoded in the lower layers, even though the initialization layer still performs worst, just as with WSD, demonstrating that it is not sufficient by itself.

These empirical results suggest that any layer pooling strategy based on a fixed set of layers, such as the often used sum of layers [-1,-4], cannot accurately capture the available sense information encoded in pre-trained NLMs.

Word Sense Disambiguation (WSD)



Uninformed Sense Matching (USM)

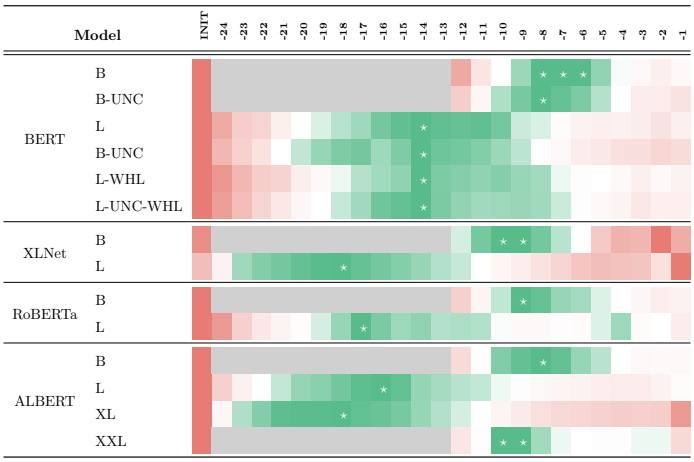


Table 3: Performance variation on the development set across all models and configurations considered in this work. Green represents best performing layers (best is marked with ⋆), red represents worst performing layers, and grey stands for layers missing in shallower variants.

6.2. Sense Profiles for Disambiguation and Matching

In Section 4.4.2 we described our method for uncovering a model-specific set of layer weights which informs a weighted layer pooling that results in improved sense representations. We have applied this method to all our models, for both disambiguation and matching scenarios, in order to verify whether our proposed method reliably improves performance on WSD and USM tasks when compared to conventional pooling approaches. Additionally, we also compare against different values for the temperature t parameter. In order to understand whether our recommended t values actually result in improved test-time performance for these tasks, we run a limited evaluation on the ALL test set of Raganato et al. (2017) where we compare NLMs using only our method as described until Section 4.2.1 (without using glosses) and trained solely with SemCor annotations. Later, in Sections 7.1 and 7.2, we report WSD and USM results using our final solution in comparison with the current state-of-the-art.

The conventional layer choices we considered are the following: last/final (L_{-1}); second-to-last (L_{-2}); sum of last 4 ($L_{-1} + L_{-2} + L_{-3} + L_{-4}$); integer weighted sum of last 4 ($L_{-1} + 2 * L_{-2} + 3 * L_{-3} + 4 * L_{-4}$); fractional weighted sum of last 4 ($\frac{1}{4} * L_{-1} + \frac{1}{3} * L_{-2} + \frac{1}{2} * L_{-3} + L_{-4}$). We tested temperature values $t \in \{0.002, 0.005, 0.01, 0.1, 1.0\}$[6]. Below we discuss our findings regarding the impact of sense profiles specific to each task.

The WSD validation results on Table 4 reveal that the single best layer (which varies depending on model, see Table 3) consistently outperforms the sum of last 4 layers. We also find that WSD sense profiles with $t = 0.002$ and $t = 0.005$ perform comparably to the single best layer, with $t = 0.002$ being slightly closer on average. Given the close performance, we opt for recommending $t = 0.005$ as higher values are less likely to overfit on the validation set (bias-variance tradeoff). In the limited evaluation results on Table 5 we compare the performance of conventional layer choices against WSD sense profiles with the recommended temperature value. We observe that for 11 out of 14 models,

[6] $t = 0.001$ results in large exponents that cause overflow errors.

WSD sense profiles with the recommended temperature reliably outperform any of the conventional choices, of which none stands out as a reliable cross-model choice. Moreover, on Table 5 we also see that WSD sense profiles with the recommended temperature generally match or outperform both the single layers which performed best on the validation set, and the WSD sense profiles using the temperature value that showed best performance on the validation set.

Our findings regarding performance of USM sense profiles largely follow the previously mentioned findings for WSD sense profiles. In the case of USM, validation results on Table 6 more clearly show that $t = 0.100$ performs better, although $t = 1.000$ also performs well. As for the limited evaluation results, Table 7 shows that conventional layer choices significantly underperform any of the alternatives introduced in this work, with the USM sense profile with recommended temperature ($t = 0.100$) showing overall best performance.

6.3. Choosing Sense Profiles for Different Tasks

Having established that our proposed sense profiles improve WSD and USM performance over conventional layer choices, the question remains of whether to choose WSD or USM sense profiles to represent sense embeddings. In this work we propose choosing sense profiles based on the probing task that shares most similar constraints to the downstream task of interest. More specifically, tasks requiring comparison of different senses for the same word fit the disambiguation profile, such as classical WSD (Navigli, 2009) or WiC (Pilehvar & Camacho-Collados, 2019), and benefit less from information in lower layers. On the other hand, tasks without lexical constraints, not only USM but also synset similarity (Colla et al., 2020a) or semantic change (Hamilton et al., 2016), are better suited to the matching profile, which uses information from more layers. In Section 7 we evaluate sense embeddings learned using sense profiles according to each task's constraints, and in Section 8.1 we analyse the performance gap when using alternate sense profiles.

Model		Sum	Layer	Weighted Sum (WS)				
		LST4	Best	t=0.002	t=0.005	t=0.010	t=0.100	t=1.000
BERT	B	71.6	**72.5** (-2)	72.4	72.2	71.8	69.6	68.6
	B-UNC	71.9	**73.0** (-2)	72.9	72.8	72.3	70.3	69.1
	L	73.8	**74.7** (-2)	**74.7**	74.5	74.3	72.2	70.9
	L-UNC	72.7	**72.9** (-2)	**72.9**	72.7	72.7	71.5	70.8
	L-WHL	72.0	73.0 (-2)	**73.1**	72.6	71.8	70.4	69.1
	L-UNC-WHL	71.5	**72.7** (-8)	72.6	72.1	72.0	71.5	70.7
XLNet	B	66.6	70.9 (-3)	71.0	**71.2**	**71.2**	70.3	69.9
	L	66.5	72.7 (-17)	73.0	**73.4**	73.3	72.4	71.8
RoBERTa	B	72.5	**72.9** (-3)	72.8	72.6	72.2	71.6	71.2
	L	74.1	**74.9** (-10)	**74.9**	74.7	74.4	73.6	73.1
ALBERT	B	68.3	**68.9** (-5)	68.3	68.2	68.1	67.6	67.3
	L	69.4	**70.5** (-15)	70.2	70.0	69.9	69.3	69.3
	XL	68.2	**71.4** (-13)	71.1	71.1	71.0	70.5	70.4
	XXL	72.4	**73.8** (-6)	73.5	73.4	73.3	72.8	72.4

Table 4: WSD validation results (F1). Reports best single layer and weighted sums using specific sense profiles, with different t values, for each model configuration. Sense representations for this experiment were learned from SemCor (no propagation required).

Model		Standard					Proposed		
		Layer -1	Layer -2	Sum LST4	WS (I) LST4	WS (F) LST4	Layer Best Dev	WS Rec. t	WS Best t
BERT	B	72.1	**72.9**	72.6	72.5	72.5	**72.9** (-2)	72.8	**72.9** (.002)
	B-UNC	**73.5**	**73.5**	73.0	73.3	73.3	**73.5** (-2)	73.4	**73.5** (.002)
	L	73.3	73.9	74.0	74.0	74.0	73.9 (-2)	**74.2**	74.0 (.002)
	L-UNC	73.4	73.6	73.9	**74.0**	**74.0**	73.6 (-2)	**74.0**	73.8 (.002)
	L-WHL	72.0	73.4	73.2	73.1	73.0	73.4 (-2)	**73.5**	73.4 (.002)
	L-UNC-WHL	72.0	73.0	72.9	72.8	72.8	65.4 (-8)	**73.1**	**73.1** (.002)
XLNet	B	69.1	67.4	64.8	62.7	63.7	55.4 (-3)	**72.3**	**72.3** (.005)
	L	66.2	70.4	65.7	64.8	66.1	57.5 (-17)	**73.8**	**73.8** (.005)
RoBERTa	B	71.9	73.3	73.3	73.4	73.3	73.5 (-3)	73.6	**73.7** (.002)
	L	71.2	74.0	74.1	74.0	73.9	66.3 (-10)	**74.7**	**74.7** (.002)
ALBERT	B	**70.6**	69.6	70.1	70.1	70.3	67.3 (-5)	69.7	69.7 (.002)
	L	70.1	70.5	70.5	70.6	70.4	67.7 (-15)	**71.1**	70.7 (.002)
	XL	64.3	69.0	68.8	67.8	67.2	66.6 (-12)	**73.0**	**73.0** (.002)
	XXL	69.4	73.7	73.9	73.1	72.5	74.8 (-6)	**75.1**	**75.1** (.002)

Table 5: WSD test results (F1 on ALL). Reports conventional layer choices and alternatives using sense profiles. Recommended t for WSD is 0.005. Sense representations for this experiment were learned from SemCor (with propagation).

Model		Sum	Layer	Weighted Sum (WS)				
		LST4	Best	t=0.002	t=0.005	t=0.010	t=0.100	t=1.000
BERT	B	59.2	61.2 (-6)	61.5	61.5	61.8	**62.2**	62.0
	B-UNC	58.6	63.1 (-8)	63.1	63.2	63.5	**63.7**	**63.7**
	L	57.7	63.8 (-14)	63.9	64.0	63.9	**64.5**	**64.5**
	L-UNC	56.8	64.3 (-14)	64.4	64.6	64.6	65.4	**65.7**
	L-WHL	59.7	62.1 (-14)	62.1	62.0	62.1	**62.7**	62.5
	L-UNC-WHL	59.1	**64.8** (-14)	64.7	64.5	64.5	64.7	**64.8**
XLNet	B	34.7	61.2 (-9)	61.3	61.4	61.4	**61.9**	60.4
	L	28.0	63.6 (-18)	63.5	63.6	63.7	**64.4**	64.1
RoBERTa	B	61.6	**64.2** (-9)	64.0	63.9	64.0	64.0	63.9
	L	64.1	65.3 (-17)	65.2	65.4	65.8	66.1	**66.2**
ALBERT	B	60.0	61.8 (-8)	**61.9**	61.8	61.7	61.6	61.6
	L	60.0	**64.1** (-16)	63.5	63.5	63.2	63.3	63.2
	XL	54.9	64.5 (-18)	64.2	64.1	64.2	64.5	**64.6**
	XXL	65.8	65.8 (-9)	65.7	66.1	66.1	66.2	**66.3**

Table 6: USM validation results (F1). Reports best single layer and weighted sums using specific sense profiles, with different t values, for each model configuration. Sense representations for this experiment were learned from SemCor (no propagation required).

Model		Standard					Proposed		
		Layer	Layer	Sum	WS (I)	WS (F)	Layer	WS	WS
		-1	-2	LST4	LST4	LST4	Best Dev	Rec. t	Best t
BERT	B	53.1	51.2	53.7	53.0	53.0	57.0 (-6)	**57.7**	**57.7** (.100)
	B-UNC	50.4	50.4	53.0	51.8	52.3	57.9 (-8)	**58.8**	**58.8** (.100)
	L	53.9	50.3	52.5	52.8	53.4	58.9 (-14)	**60.0**	**60.0** (.100)
	L-UNC	48.3	46.7	49.0	48.8	48.8	58.5 (-14)	60.3	**60.4** (1.00)
	L-WHL	53.3	54.3	54.5	54.2	54.5	57.6 (-14)	**58.4**	**58.4** (.100)
	L-UNC-WHL	53.7	52.4	53.3	53.3	53.3	58.3 (-14)	59.4	**59.6** (1.00)
XLNet	B	38.1	36.9	31.4	27.8	28.9	**57.3** (-9)	**57.3**	**57.3** (.100)
	L	27.9	41.3	28.7	28.2	29.6	59.0 (-18)	**60.4**	**60.4** (.100)
RoBERTa	B	53.7	53.2	55.1	55.3	55.5	58.3 (-9)	**59.2**	**59.2** (.100)
	L	56.3	56.7	57.9	58.1	58.0	60.6 (-17)	**61.2**	**61.2** (.100)
ALBERT	B	53.8	53.6	54.7	54.2	54.8	55.8 (-8)	56.2	**56.3** (.002)
	L	55.7	55.5	56.0	56.1	56.1	57.4 (-16)	**58.4**	57.3 (.005)
	XL	41.2	48.5	49.8	48.0	47.4	59.9 (-18)	**60.1**	59.8 (1.00)
	XXL	55.1	60.6	61.7	61.0	60.5	60.6 (-9)	**62.3**	**62.3** (1.00)

Table 7: USM test results (F1 on ALL). Reports conventional layer choices and alternatives using sense profiles. Recommended t for USM is 0.1. Sense representations for this experiment were learned from SemCor (with propagation).

7. Evaluation

In this work we address several sense-related tasks selected to investigate the versatility of the proposed sense embeddings, covering disambiguation (§7.1 - WSD), matching (§7.2 - USM), meaning change detection (§7.3 and §7.4 - WiC and GWCS) and sense similarity (§7.5 - SID). For each task, we report our new results (LMMS-SP) in comparison with the state-of-the-art and the original LMMS (2019a) sense embeddings.

For brevity, we only consider the variant from each model family that showed best results in our probing analysis (§6). In our comparisons, we omit LMMS_{2348} because those sense embeddings are concatenated with fastText (Bojanowski et al., 2017) word embeddings, thus not exclusively based on representations from particular NLMs, as focused in this work.

All tasks are solved essentially using cosine similarity between contextual embeddings and LMMS-SP precomputed sense embeddings represented using the same NLM. Each task's subsection provides more details about how these similarities are used to produce task-specific predictions. No additional task-specific training or validation datasets are used asides from those referred in Section 5.2, and all NLMs are employed in the same exact fashion - simply retrieving contextualized representations from each layer (following §4.1).

As such, LMMS-SP performance on these tasks should be indicative of each NLM's intrinsic ability to approximate meaning representations learned during pre-training with language modelling objectives alone.

7.1. Word Sense Disambiguation (WSD)

Sentence	**Lemma**	**POS**	**Gold Sensekey**
Eyes that were clear , but also bright with a strange intensity , a sort of cold **fire** burning behind them .	fire	NOUN	fire%1:12:00::

Table 8: Example WSD instance from Raganato et al. (2017). Sentence, lemma and part-of-speech (POS) are provided. The goal is to predict the correct sensekey (sense from WordNet).

WSD is the most popular and obvious task for evaluating sense embeddings. This task has been researched since the early days of Artificial Intelligence and constitutes an AI-complete task (Navigli, 2009). It is usually formulated as choosing the correct sense for a word in context out of a list of possible senses given the word's lemma and part-of-speech tag (see Table 8). Several test sets have been proposed over the years, and the compilation of Raganato et al. (2017) has emerged as the de facto evaluation framework for English WSD, which we also use. Naturally, this task suits sense profiles for WSD, and we follow the method described in Section 4.3.

7.1.1. Results

On Table 9 we report performance on the standard test sets of the WSD Evaluation Framework (Raganato et al., 2017). Given the breadth of recent WSD solutions, we make results more comparable by separating solutions using only SemCor annotations, and solutions augmenting SemCor with other sense-annotated datasets. In the case of LMMS-SP, we combine SemCor with the unambiguous annotations from UWA, which are easily retrieved from unlabeled corpora. We also report which solutions use glosses and relations, besides sense annotations, as well as which solutions are based on 1NN (first nearest neighbor) with precomputed sense embeddings represented in the space of NLMs.

When considering SemCor as the only source of annotations, LMMS and LMMS-SP remain the best solutions based on 1NN in NLM-space. Most notably, LMMS-SP$_{\text{ALBERT-XXL}}$ is able to match the performance of LMMS$_{2048}$ on the combination of test sets (ALL) without concatenating gloss embeddings.

As could be expected, task-specific classifiers show best results, particularly BEM (Blevins & Zettlemoyer, 2020) and ConSeC (Barba et al., 2021). Generally, solutions fine-tuning NLMs, or combining them with other classifiers trained for the WSD task show improved performance over 1NN. Despite this, in Loureiro et al. (2021) we have shown that 1NN solutions offer other advantages, such as better sample efficiency and less frequency biases, besides the versatility advocated in this current work.

Allowing for additional annotations, we find that LMMS-SP results improve slightly when using BERT-L and ALBERT-XXL. ARES (Scarlini et al., 2020b) uses semi-supervised annotations to increase coverage of the sense inventory with sense embeddings represented on the space of BERT-L. Results show the ARES dataset leads to improved WSD performance in comparison to LMMS-SP on the reported test sets, particularly on SE13 and SE15.[7]

	Model	1NN	Defs.	Rels.	SE2 (n=2,282)	SE3 (n=1,850)	SE07 (n=445)	SE13 (n=1,644)	SE15 (n=1,022)	ALL (n=7,253)
SemCor (SC)	MFS				65.6	66.0	54.5	63.8	67.1	64.8
	context2vec (2016)	✓			71.8	69.1	61.3	65.6	71.9	69.0
	ELMo (2018a)	✓			71.6	69.6	62.2	66.2	71.3	69.0
	BERT-L (2019a)	✓			76.3	73.2	66.2	71.7	74.1	73.5
	SVC (2019)			✓	76.6	76.9	69.0	73.8	75.4	75.4
	GlossBERT (2019)		✓		77.7	75.2	72.5*	76.1	80.4	77.0*
	EWISER (2020)		✓	✓	78.9	78.4	71.0	78.9	79.3*	78.3*
	BEM (2020)		✓		79.4	77.4	74.5*	79.7	81.7	79.0*
	ConSeC (2021)		✓		82.3	79.9	77.4*	83.2	85.2	82.0*
	LMMS_{1024} (2019a)	✓		✓	75.4	74.0	66.4	72.7	75.3	73.8
	LMMS_{2048} (2019a)	✓	✓	✓	76.3	<u>75.6</u>	68.1	75.1	77.0	<u>75.4</u>
	$\text{LMMS-SP}_{\text{BERT-L}}$	✓	✓	✓	76.1	74.0	67.0	75.2	<u>77.4</u>	75.0
	$\text{LMMS-SP}_{\text{XLNet-L}}$	✓	✓	✓	76.0	73.1	66.4	74.2	74.9	74.1
	$\text{LMMS-SP}_{\text{RoBERTa-L}}$	✓	✓	✓	77.2	73.5	67.9	<u>75.5</u>	76.4	75.2
	$\text{LMMS-SP}_{\text{ALBERT-XXL}}$	✓	✓	✓	<u>77.4</u>	74.8	<u>71.0</u>	74.7	74.8	<u>75.4</u>
SC+Others	SVC (2019)			✓	79.4	78.1	71.4	77.8	81.4	78.5
	KnowBERT (2019)			✓	76.4	76.0	71.4	73.1	75.4	75.1
	EWISER (2020)		✓	✓	80.8	79.0	75.2	80.7	81.8*	80.1*
	ARES (2020b)	✓	✓		78.0	77.1	71.0	77.3	83.2	77.9
SC+UWA	$\text{LMMS-SP}_{\text{BERT-L}}$	✓	✓	✓	76.7	74.1	66.4	75.2	<u>77.6</u>	75.2
	$\text{LMMS-SP}_{\text{XLNet-L}}$	✓	✓	✓	76.1	73.1	65.9	74.2	75.0	74.1
	$\text{LMMS-SP}_{\text{RoBERTa-L}}$	✓	✓	✓	77.4	73.5	67.7	<u>75.3</u>	76.7	75.2
	$\text{LMMS-SP}_{\text{ALBERT-XXL}}$	✓	✓	✓	<u>77.7</u>	<u>75.0</u>	<u>70.5</u>	74.7	74.9	<u>75.5</u>

Table 9: F1 scores (%) for each test set in the WSD Evaluation Framework (Raganato et al., 2017). Top rows show results for models using sense annotations exclusively from SemCor (SC). Bottom rows show results for models augmenting SC with annotations from additional sources. Results marked with * correspond to development sets (and therefore ALL). For each group of results, we underline the best from LMMS.

[7] We expect the annotations in ARES to produce further performance gains for LMMS-SP but do not use this resource due to its large size (13x the annotations in SemCor+UWA10).

7.2. Uninformed Sense Matching (USM)

Sentence	**Gold Sensekey**	**Gold Synset**
Eyes that were clear , but also bright with a strange intensity , a sort of cold **fire** burning behind them .	fire%1:12:00::	06711159n (fire^{9}_{n})

Table 10: Example USM instance adapted from Raganato et al. (2017). The correct sensekey or synset must be predicted, in separate evaluations.

We introduced the USM task in Loureiro & Jorge (2019a) as a variation on WSD that can more accurately represent the extent to which NLMs can associate words or phrases to senses from the WordNet inventory. The crucial difference in relation to WSD is that in the USM task we do not use any supplemental information to restrict candidates in the sense inventory (compare examples in Table 8 and Table 10). Conveniently, this allows for USM to use the same test sets as WSD. As expected, we address USM using the sense profile of the same name, and follow the method described in Section 4.3. In this work we evaluate USM from both the sensekey and synset perspective, to provide a clearer account of the impact of lexical information on task performance.

7.2.1. Results

Following Loureiro & Camacho-Collados (2020), we evaluate performance considering two additional metrics besides F1: Precision at 5 (P@5) and Mean Reciprocal Rank (MRR). To our knowledge, ARES (Scarlini et al., 2020b) is the only other publicly available set of full-coverage sense embeddings represented in the space of a Transformer-based NLM, so we also compare LMMS-SP against those sense embeddings. Since our prior LMMS sense embeddings and the ARES sense embeddings are released using sensekey representations, USM synset evaluation requires converting those sensekey embeddings to synset embeddings. We perform this conversion by simply averaging sensekey embeddings that belong to the same synset. In Section 8.2 we analyse the impact this conversion can have on task performance.

On Table 11 it can be observed that LMMS-SP dramatically improves performance over LMMS on all three metrics considered. The poor performance of LMMS_{2048} in comparison to LMMS_{1024} suggests that concatenating gloss embeddings is detrimental to USM performance, particularly on the F1 metric. In this comparison we do not consider LMMS_{2348} because those sense embeddings are concatenated with fastText static embeddings, resulting in 300 dimensions having the same exact distribution for sense embeddings corresponding to identical lemmas. This property of LMMS_{2348} makes the comparison inequitable and diverts from this work's focus on the intrinsic capabilities of Transformer NLMs.

Interestingly, we find that, when targeting sensekeys, $\text{LMMS-SP}_{\text{ALBERT-XXL}}$ shows best performance on all metrics, and ARES (based on BERT-L) only outperforms $\text{LMMS-SP}_{\text{BERT-L}}$ on the F1 metric. However, when targeting synsets, the additional contexts of ARES prove more advantageous, and we do not observe a similar performance gap between sensekeys and synset as we do with LMMS-SP, which can be expected considering that the additional contexts of ARES are targeted at the synset-level.

Model	Sensekeys			Synsets		
	F1	**P@5**	**MRR**	**F1**	**P@5**	**MRR**
ARES	61.4	84.7	71.8	**60.7**†	**86.5**†	**71.8**†
LMMS_{1024} (2019a)	52.2	66.9	59.0	29.4†	53.9†	40.7†
LMMS_{2048} (2019a)	34.8	60.3	46.3	32.5†	58.9†	44.5†
$\text{LMMS-SP}_{\text{BERT-L}}$	60.8	86.7	72.2	51.0	81.7	64.3
$\text{LMMS-SP}_{\text{XLNet-L}}$	60.1	87.3	71.9	51.7	82.7	65.1
$\text{LMMS-SP}_{\text{RoBERTa-L}}$	62.2	86.9	73.1	50.2	80.1	63.3
$\text{LMMS-SP}_{\text{ALBERT-XXL}}$	**62.9**	**87.6**	**73.7**	52.7	81.9	65.5

Table 11: USM results on the ALL test set of the WSD Evaluation Framework (Raganato et al., 2017), at sense and synset-level. Results marked with † are obtained from synset embeddings converted from sensekey embeddings.

Sentence Pairs	Lemma	POS	Boolean
You must **carry** your camping gear . Sound **carries** well over water .	carry	VERB	False
He wore a jock strap with a metal **cup** . Bees filled the waxen **cups** with honey .	cup	NOUN	True

Table 12: Examples from the WiC training set. Showing two independent instances.

7.3. Word-in-Context (WiC)

The Word-in-Context (Pilehvar & Camacho-Collados, 2019, WiC) task is designed to assess how context impacts word representations produced by contextual NLMs. It is a binary classification task that simply requires determining whether a particular word is used with the same meaning or not in a pair of sentences, also given lemma and POS provided in WSD tasks (see Table 12 for examples). The dataset is balanced and performance is measured with accuracy.

7.3.1. Solution

In this work, we tackle the WiC task using our proposed sense embeddings following the unsupervised approach from Loureiro & Jorge (2019b), which essentially applies the 1NN method for disambiguating the target word in both sentences and checks whether they are equal or not. Even though we also explored a supervised approach in Loureiro & Jorge (2019b), based on Logistic Regression, in this work we focus on the unsupervised approach as its performance is more revealing of the inherent representational abilities of NLMs. Given the close relation to disambiguation, we use WSD sense profiles for WiC.

7.3.2. Results

WiC is a benchmark NLU task, being part of SuperGLUE (Wang et al., 2019), therefore most state-of-the-art NLMs have reported results for this task. The initial baseline methods proposed with WiC were based on cosine similarity with thresholds learned from the validation set.

Most recent solutions, however, involve fine-tuning the NLM (as performed for other sentence classification tasks in SuperGLUE) using the training and validation sets provided with WiC. One notable exception is Scarlini et al. (2020b) which proposed a method that leverages ARES sense embeddings to improve the fine-tuning process. As such, on Table 13 we compare results from these solutions to our unsupervised LMMS and LMMS-SP, as well as an unsupervised result based on the same 1NN approach using the ARES embeddings.

Starting with our unsupervised results, we confirm that LMMS-SP$_{\text{BERT-L}}$ surpasses the performance of LMMS$_{2048}$ (based on BERT-L), and once again LMMS-SP$_{\text{ALBERT-XXL}}$ displays the best performance. Nevertheless, supervised solutions using NLMs fine-tuned for this task show best performance overall, particularly T5 (Raffel et al., 2020) which is currently the largest NLM with reported results on this task, at over 11B parameters. KnowBERT (Peters et al., 2019) and SenseBERT (Levine et al., 2020) are both NLMs based on BERT that have been augmented with sense information from WordNet and SemCor, among other resources, showing improved performance in comparison to fine-tuning the original BERT-L.

The method used by Scarlini et al. (2020b) to employ sense embeddings while fine-tuning BERT-L for WiC resulted in a notable improvement similar to SenseBERT. In the unsupervised setting, however, we found that ARES embeddings outperform LMMS-SP$_{\text{BERT-L}}$, but underperform both LMMS-SP$_{\text{RoBERTa-L}}$ and LMMS-SP$_{\text{ALBERT-XXL}}$. We expect following the same method to assist supervised fine-tuning with LMMS-SP sense embeddings may produce improved results, but consider that experiment out of scope for this work.

As for solutions using the threshold method, all reported models substantially underperform unsupervised results using any Transformer-based NLM.

7.4. Graded Word Similarity in Context (GWCS)

For evaluating graded contextual similarity, in contrast to the binary contextual similarity assignments of WiC, we address SemEval 2020 Task 3: Graded Word Similarity in Context (GWCS) (Armendariz et al., 2020a). This task,

	Method	Language Model	Sense Embeddings	Acc.
Supervised	Fine-Tuning	BERT-L (2019)	-	69.6*
	Logistic Reg.	BERT-L	$LMMS_{2048}$ (2019b)	68.1
	Fine-Tuning	RoBERTa-L (2019b)	-	69.9*
	Fine-Tuning	KnowBERT (2019)	-	70.9
	Fine-Tuning	SenseBERT (2020)	-	72.1
	Fine-Tuning	T5 (2020)	-	**76.9***
	Fine-Tuning	BERT-L	ARES (2020b)	72.2*
Threshold	1NN WSD	-	JBT (2016)	53.6
	1NN WSD	-	DeConf (2016)	58.7
	1NN WSD	context2vec (2016)	-	**59.3**
	1NN WSD	-	SW2V (2017)	58.1
	1NN WSD	ELMo (2018a)	-	57.7
	1NN WSD	-	LessLex (2020b)	59.2
Unsupervised	1NN WSD	BERT-L	ARES (2020)	67.6
	1NN WSD	BERT-L	$LMMS_{2048}$ (2019a)	66.3
	1NN WSD	BERT-L	$LMMS\text{-}SP_{BERT\text{-}L}$	67.4
	1NN WSD	XLNet-L	$LMMS\text{-}SP_{XLNet\text{-}L}$	66.1
	1NN WSD	RoBERTa-L	$LMMS\text{-}SP_{RoBERTa\text{-}L}$	67.8
	1NN WSD	ALBERT-XXL	$LMMS\text{-}SP_{ALBERT\text{-}XXL}$	**67.9**

Table 13: Results (Accuracy) on the test set of the WiC task comparing our unsupervised approach to the state-of-art. Best results for each approach reported in bold. Our results were obtained from the Codalab online platform. Results marked with * used the SuperGLUE version of the WiC test set, which has minor preprocessing differences.

based on the CoSimLex resource (Armendariz et al., 2020b), targets word pairs used for evaluating distributional semantic models (not necessarily polysemous words) in contexts spanning multiple sentences. The task is divided into two sub-tasks derived from human-annotated similarity ratings: 1) predict the change in similarity between two different contexts for each word pair; 2) predict the similarity ratings themselves. Table 14 shows a single example from GWCS, featuring two contexts each with occurrences of the same pair of words, context specific similarity ratings, and the associated similarity change.

	Contexts	Sim.	Change
A	Tim Drake **keeps** a memorial for her in his cave hideout underneath Titans Tower in San Francisco. [...] It is later revealed that Dr. Leslie Thompkins had faked her death after the gang war in an effort to **protect** her.	4.44	-0.52
B	Shisa are wards, believed to **protect** from various evils. When found in pairs, the shisa on the left traditionally [...] The open mouth to ward off evil spirits, and the closed mouth to **keep** good spirits in.	3.92	

Table 14: Example from the practice set of GWCS (single instance). Contexts A and B each have corresponding similarity ratings for the same 'keep'-'protect' word pair.

7.4.1. Solution

While the sub-tasks are independently evaluated, we employ essentially the same method for both, based on our straightforward approach for the WiC task covered in Section 7.3, with minor adjustments to quantify the observed change in similarity. Given contexts A and B, we disambiguate target words (each instance's word pair) in the corresponding contexts, and compute sense similarities sim^A_{wsd} and sim^B_{wsd} as the cosine similarity between the embeddings of the predicted senses. Considering that disambiguation may predict the same senses, thus resulting in $sim^A_{wsd} = sim^B_{wsd}$ for many instances, we also compute contextual similarities sim^A_{ctx} and sim^B_{ctx} as the cosine similarity between the contextual embeddings of the target words. Thus, we determine similarity scores specific to context A as $sim^A = \frac{1}{2}(sim^A_{wsd} + sim^A_{ctx})$, and similarity scores specific to context B as $sim^B = \frac{1}{2}(sim^B_{wsd} + sim^B_{ctx})$. These context-specific similarities constitute our solutions to sub-task 2. We determine the semantic change scores for sub-task 1 trivially as $sim^B - sim^A$. Considering that this solution closely follows our solution for WiC, and that word pairs contained in this dataset tend to be closely related, we use the WSD sense profile for GWCS.

7.4.2. Results

Performance on sub-task 1 is measured with Pearson Uncentered Correlation between the system's scores and the average human annotations, and perfor-

mance on sub-task 2 is measured with the harmonic mean of the Spearman and Pearson correlations between the system's scores and the average human annotations. On Table 15 we report results using our sense embeddings (LMMS and LMMS-SP), using the ARES sense embeddings with our scoring method (and BERT-Large, pooling with the sum of last 4 layers), and the best reported results from other task participants (including post-evaluation, until 03/2021). Similarly to WiC, the scores for the test sets are hidden from participants, both during evaluation (ended 03/2020) and post-evaluation periods (extends indefinitely), so all reported results are obtained from the online platform used by SemEval after submitting each system's predictions.

We observe that our straightforward method combining similarity between sense and contextual embeddings is able to outperform the solutions of other task participants (Leaderboard Best), most of which also relied on Transformer-based NLMs (Armendariz et al., 2020a). Interestingly, GWCS shows wide variation in performance from the choice of NLM, with LMMS-SP$_{\text{XLNet-L}}$ standing out with clearly best results on both sub-tasks[8].

Model	**Subtask1**	**Subtask2**
Leaderboard Best†	77.4[1]	74.6[2]
ARES (2020b)	76.9	74.5
LMMS$_{1024}$ (2019a)	74.1	74.2
LMMS$_{2048}$ (2019a)	75.7	74.5
LMMS-SP$_{\text{BERT-L}}$	76.2	74.4
LMMS-SP$_{\text{XLNet-L}}$	**78.7**	**76.6**
LMMS-SP$_{\text{RoBERTa-L}}$	75.7	74.9
LMMS-SP$_{\text{ALBERT-XXL}}$	75.2	71.8

Table 15: Results on both subtasks of SemEval 2020 Task 3. † as of 03/2021, considering evaluation and post-evaluation submissions (Users: [1]Ferryman, [2]Alexa). ARES results obtained using the same method as LMMS, only replacing the corresponding sense embeddings.

[8]Complete leaderboard results are available on Appendix A. Additionally, Appendix B reports performance on the Stanford Contextual Word Similarities (Huang et al., 2012) task, which inspired the GWCS and WiC tasks, with similar conclusions.

7.5. Sense Similarity

Synset 1	Synset 2	Similarity
08570634n (hayfield^1_n)	08598301n (grassland^1_n)	3.58
03169390n (decoration^1_n)	03291741n (envelope^2_n)	0.08

Table 16: Two examples of paired synsets with human similarity ratings from the SID dataset. Showing synset identifiers after conversion to WordNet (more readable format in parenthesis).

All the tasks we considered so far (WSD, USM, WiC and GWCS) have evaluated sense embeddings by their utility for accurately matching or distinguishing word senses in particular contexts. In this last task, we address intrinsic evaluation of sense embeddings, directly comparing distributional similarity between sense pairs against human similarity ratings.

We perform this evaluation using the Sense Identification Dataset (Colla et al., 2020c, SID), which is based on the word pairs (nouns only) and human similarity ratings from SemEval-2017 Task 2 (Camacho-Collados et al., 2017), with the addition of mapping word pairs to particular senses in the BabelNet sense inventory (see examples on Table 16).

7.5.1. Task Adaptation

We convert the BabelNet sense identifiers to synsets from WordNet 3.0 using the mapping provided by Navigli & Ponzetto (2010). However, some instances cannot be mapped due to missing entries in WordNet, or, in rare cases, their mapping results in the two senses of the pair being equal, leading to a reduction of 492 instances to 377 mapped to WordNet. We further split SID into different groups for additional insights. We first separate the 354 pairs for which both senses are represented in the related works we compare against (overlapping), considering these are not always complete sets of WordNet sense embeddings. Next, we breakdown the overlapping pairs into a set of the most polarized word pairs (i.e., similarity ratings ≤ 1 or ≥ 3), and another set containing only pairs with senses that are annotated in SemCor+UWA10 (observed).

7.5.2. Solution

We use cosine similarity between synset embeddings to correlate with human similarity ratings. Since we are directly comparing embeddings of very different dimensionality, we apply truncated SVD to normalize them to 300 dimensions[9] (including related work). The senses being compared range from completely unrelated (e.g., polyhedron$_n^1$; actor$_n^1$) to highly related or similar (e.g., actor$_n^1$; actress$_n^1$), so we use USM sense profiles for SID.

7.5.3. Results

Performance on SID is measured with Pearson correlation. For completeness, we report performance of synset embeddings that are not based on contextual NLMs, including new results based on fastText embeddings (trained on CommonCrawl)[10]. Results for related works are based on sense embeddings provided by the authors, converting sensekeys to synsets by averaging the corresponding embeddings whenever required (as in Section 7.2). The inter-annotator agreement on our full set (n=377) reaches 87.9, measured as averaged pairwise Pearson correlation of the original SemEval-2017 human similarity scores.

Results for the WordNet-subset of SID are shown on Table 17. As can be observed, LMMS-SP substantially outperforms LMMS and related works. As with GWCS, LMMS-SP$_{\text{XLNet-L}}$ stands out with clearly best results. While LMMS-SP also outperforms most non-contextual embeddings, it still underperforms LessLex (Colla et al., 2020b) embeddings, which are based on ensembles and learned using BabelNet. We also note that LMMS-SP performs particularly well on the 'Observed' set corresponding to senses learned from annotated corpora. The performance gap between ARES and LMMS-SP$_{\text{BERT-L}}$ suggests that additional semi-supervised annotations for more senses may not suffice. Finally, the 'Polarized' set seems consistently easier than the full set, indicating that the most challenging pairs are those with moderate similarity ratings.

[9] We verified that SVD-reduced embeddings always outperform original embeddings.

[10] fastText embeddings for a given synset are computed by averaging the word embeddings for each lemma that belongs to the input synset in WordNet.

	Synset Embeddings	WN Full Coverage	All (n=377)	Overlapping All (n=354)	Overlapping Polarized (n=182)	Overlapping Observed (n=297)
Static	fastText (2017)	✓	64.4	63.5	69.3	65.4
	$\text{NASARI}_{\text{UMBC}}$ (2015)		-	71.6	79.1	74.4
	DeConf† (2016)		75.1	74.9	80.6	76.9
	LessLex (2020b)		**82.5**	**82.3**	**85.5**	**85.1**
Contextual	SensEmBERT (2020a)		66.9	66.8	74.6	69.5
	ARES†	✓	70.6	70.4	80.5	73.3
	LMMS_{2048}† (2019a)	✓	71.2	72.2	76.2	76.3
	$\text{LMMS-SP}_{\text{BERT-L}}$	✓	77.8	77.8	80.4	83.1
	$\text{LMMS-SP}_{\text{XLNet-L}}$	✓	**79.5**	**79.6**	**81.2**	**84.5**
	$\text{LMMS-SP}_{\text{RoBERTa-L}}$	✓	74.1	74.2	79.0	80.9
	$\text{LMMS-SP}_{\text{ALBERT-XXL}}$	✓	77.4	77.2	80.5	81.4

Table 17: Performance (Pearson Correlation) on the adapted SID dataset. All reported embeddings feature 300 dimensions. Embeddings marked with † have been converted from sensekeys. LessLex and NASARI embeddings were converted from BabelNet to WordNet using the same mapping applied to the SID adaptation.

8. Analysis

In this section, we perform several ablation studies to better understand the impact of individual contributions we have introduced in this work's extension of LMMS. These experiments target the same NLMs and tasks[11] that we addressed in the previous evaluation section. Our ablation analyses cover the impact of sense profiles (§8.1), UWA annotations (§8.2), merging gloss representations (§8.3) and indirect representation of synsets (§8.4). Considering that part-of-speech is an important factor in disambiguation (and sense representation), we also report performance per part-of-speech using both LMMS and LMMS-SP sense embeddings on WSD and USM tasks (§8.5).

[11] Due to leaderboard submission limits, ablations for WiC use the validation set.

8.1. Choice of Sense Profiles

On Table 18 we report performance according to the sense profile used for weighted pooling of contextual embeddings from NLMs (described in Section 6.2), and using the sum of the last 4 layers (Sum-LST4), as commonly used in related work and the original LMMS (Loureiro & Jorge, 2019a).

Our first conclusion is that Sum-LST4 pooling is only appropriate for particular models and tasks (i.e., WSD and WiC w/BERT-L; WiC w/RoBERTa-L), but detrimental for most (specially any task w/XLNet-L; any model for USM). However, our recommended choice of sense profile not only appears beneficial for WSD and USM tasks across all models (expected since the sense profiles are based on those tasks), but also for WiC, GWCS and SID. In fact, out of 20 model-task combinations, we only find 3 exceptions: RoBERTa-L on WiC and GWCS, and ALBERT-XXL on GWCS (to a lesser extent). Moreover, we confirm that tasks are sensitive to the choice between SP-WSD and SP-USM, which validate our task-specific recommendations.

Task (Metric)	Pooling	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
WSD (F1 on ALL)	Sum-LST4	75.2	56.4	74.8	73.8
	SP-WSD ⋆	75.2	74.1	75.2	**75.5**
	SP-USM	72.9	73.4	74.2	74.5
USM (P@5 on ALL)	Sum-LST4	74.6	65.9	74.6	74.3
	SP-WSD	73.6	81.6	83.1	85.7
	SP-USM ⋆	86.7	87.3	86.9	**87.6**
WiC (ACC on Val.)	Sum-LST4	71.8	61.0	**72.1**	66.8
	SP-WSD ⋆	71.8	67.9	68.8	68.7
	SP-USM	67.7	65.0	68.5	67.9
GWCS (COR on ST2)	Sum-LST4	74.4	54.9	73.3	71.5
	SP-WSD ⋆	76.3	**78.7**	75.7	75.2
	SP-USM	73.4	75.4	77.2	75.9
SID (COR on ALL)	Sum-LST4	77.4	41.1	73.8	72.8
	SP-WSD	76.3	77.7	73.5	75.3
	SP-USM ⋆	77.8	**79.5**	74.1	77.4

Table 18: Impact of pooling operation on task performance. Underline highlights pooling operation that performed best for each NLM and task. Bold highlights NLM and pooling operation that performed best for each task. ⋆ denotes default choice of LMMS-SP.

8.2. Unambiguous Word Annotations

In this work we learnt our initial set of sense embeddings (as described in Sections 4.1 and 4.2.1) using SemCor, the only source of sense annotations used for LMMS (2019a), in combination with UWA (Loureiro & Camacho-Collados, 2020), a set of sense annotations exclusively targeting unambiguous words.

On Table 19 we present results showing the impact of UWA on task performance. As noted in Loureiro & Camacho-Collados (2020), the increase in WordNet coverage using UWA allows for disentangling dense clusters that coarsen the semantic space when relying on SemCor and network propagation alone. Consequently, we expect UWA to benefit sense matching tasks, which is confirmed by our results showing substantial improvements in USM and SID (the two tasks using SP-USM). We also find that UWA does not hinder performance on the remaining tasks for most model-task combinations (improves in most cases), with the exceptions of GWCS with RoBERTa-L and WiC with ALBERT-XXL (the former is also an exception observed in the sense profile ablation on §8.1).

As future work, we will also explore WordNet-independent procedures to discover monosemous words, such as the method introduced by Soler & Apidianaki (2021), which may lead to further improvements.

Task (Metric)	Annotations	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
WSD	SemCor	75.0	74.1	75.2	75.4
(F1 on ALL)	SemCor+UWA ⋆	75.2	74.1	75.2	**75.5**
USM	SemCor	76.3	76.5	76.1	77.4
(P@5 on ALL)	SemCor+UWA ⋆	86.7	87.3	86.9	**87.6**
WiC	SemCor	71.2	67.1	68.5	69.1
(ACC on Val.)	SemCor+UWA ⋆	**71.8**	67.9	68.8	68.7
GWCS	SemCor	76.1	**78.7**	76.3	72.7
(COR on ST2)	SemCor+UWA ⋆	76.3	**78.7**	75.7	75.2
SID	SemCor	72.1	75.2	65.3	73.5
(COR on ALL)	SemCor+UWA ⋆	77.8	**79.5**	74.1	77.4

Table 19: Impact of sense annotations on task performance. Underline highlights pooling operation that performed best for each NLM and task. Bold highlights NLM and pooling operation that performed best for each task. ⋆ denotes default choice of LMMS-SP.

8.3. Merging Gloss Representations

Another aspect of LMMS-SP that differs from LMMS (2019a) is merging gloss embeddings by averaging with sense embeddings, instead of through concatenation (described in Section 4.2.2).

Results on Table 20 reveal that concatenation only benefits WSD, with minor improvements over averaging. Alternatively, averaging shows clear improvements for all other tasks (again, the exception is GWCS w/RoBERTa-L).

We also report performance using exclusively gloss representations, and sense embeddings without gloss information. Surprisingly, these two sets of results are very close on WiC, GWCS and SID, showing that unsupervised representations learned from glosses can be competitive on particular tasks.

Task (Metric)	Glosses	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
WSD (F1 on ALL)	Without	74.6	74.3	75.3	**75.5**
	Exclusively	57.1	55.2	55.1	54.3
	Averaged ⋆	75.2	74.1	75.2	**75.5**
	Concatenated	**75.5**	74.3	75.3	74.8
USM (P@5 on ALL)	Without	83.5	83.9	83.7	84.2
	Exclusively	46.4	44.6	40.5	43.4
	Averaged ⋆	86.7	87.3	86.9	**87.6**
	Concatenated	85.0	85.7	85.8	86.1
WiC (ACC on Val.)	Without	66.8	64.6	68.7	67.1
	Exclusively	66.3	62.2	66.8	64.1
	Averaged ⋆	**71.8**	67.9	68.8	68.7
	Concatenated	69.3	66.5	68.5	67.7
GWCS (COR on ST2)	Without	75.6	77.3	75.0	74.4
	Exclusively	75.1	72.9	75.0	68.6
	Averaged ⋆	76.3	**78.7**	75.7	75.2
	Concatenated	75.7	77.2	75.8	74.8
SID (COR on ALL)	Without	69.5	72.5	62.0	68.3
	Exclusively	68.3	70.0	65.1	65.9
	Averaged ⋆	77.8	**79.5**	74.1	77.4
	Concatenated	76.7	77.9	69.9	73.8

Table 20: Impact of merging gloss representations on task performance. Underline highlights pooling operation that performed best for each NLM and task. Bold highlights NLM and pooling operation that performed best for each task. ⋆ denotes default choice of LMMS-SP.

8.4. Learning Synsets Directly

The SID task, as well as the synset version of USM, require synset-level embeddings. In Section 4.1, we explain that LMMS-SP synset embeddings are learned directly from sensekey annotations that are converted to synsets. However, in our evaluation we compare LMMS-SP with other works that are only available as sensekey embeddings, so we converted these representations into synset embeddings learned as the average of corresponding sensekey embeddings (i.e., learned indirectly).

On Table 21 we compare LMMS-SP embeddings learned directly and indirectly, showing that learning these representations directly leads to an average improvement across models of 7.3% on USM, and 1.5% on SID. The fact that indirect representation of synsets has a reduced impact on SID performance, in comparison to USM, suggests that indirect representation leads to more intermingled synset embeddings (i.e., harder to rank), but nearly as globally coherent as those learned from direct representation.

Task (Metric)	Synset Repr.	BERT-L	XLNet-L	RoBERTa-L	ALBERT-XXL
USM	Indirect	74.5	76.1	77.3	76.3
(P@5 on ALL)	Direct ⋆	81.7	**82.7**	80.1	81.9
SID	Indirect	77.3	78.6	73.0	75.5
(COR on ALL)	Direct ⋆	77.8	**79.5**	74.1	77.4

Table 21: Impact of learning synset representations directly from annotations, or indirectly as the average of corresponding sensekey embeddings. Underline highlights pooling operation that performed best for each NLM and task. Bold highlights NLM and pooling operation that performed best for each task. ⋆ denotes default choice of LMMS-SP.

8.5. Part-of-Speech Performance

In Loureiro & Jorge (2019a) we presented an error analysis targeting part-of-speech mismatch between predicted and ground-truth senses, which showed that verbs were particularly challenging. In this work, we complement those results by reporting performance by part-of-speech, while comparing LMMS (2019a) with LMMS-SP.

Our results on Table 22 confirm that verbs remain the most challenging part-of-speech to disambiguate correctly, although ALBERT-XXL shows appreciably better verb results than the other NLMs used in this work. Considering ranked USM matches, however, we find a much narrower gap between verbs and other parts-of-speech using LMMS-SP, with verbs performing comparably with adjectives and nouns, and only BERT-L showing differences larger than 1%.

It is also interesting to note that XLNet-L outperforms or equals ALBERT-XXL on USM for all parts-of-speech with the exception of adverbs, providing better insight into the overall performance differences reported in USM evaluation (§7.2), where ALBERT-XXL outperforms XLNet-L.

Model	Nouns		Verbs		Adjectives		Adverbs	
	WSD	**USM**	**WSD**	**USM**	**WSD**	**USM**	**WSD**	**USM**
MFS	67.6	N/A	49.6	N/A	78.3	N/A	80.5	N/A
LMMS_{1024} (2019)	75.6	48.2	63.6	65.3	79.8	75.6	**85.0**	78.6
LMMS_{2048} (2019)	78.0	54.3	64.0	64.6	**80.7**	74.0	83.5	77.2
$\text{LMMS-SP}_{\text{BERT-L}}$	78.0	87.2	63.0	84.2	80.3	85.3	83.8	96.5
$\text{LMMS-SP}_{\text{XLNet-L}}$	76.8	**87.5**	63.3	86.6	76.9	**86.8**	**85.0**	96.5
$\text{LMMS-SP}_{\text{RoBERTa-L}}$	**78.2**	86.9	63.1	**86.8**	79.2	86.1	84.7	96.2
$\text{LMMS-SP}_{\text{ALBERT-XXL}}$	77.8	87.3	**65.6**	86.6	79.1	86.7	84.1	**97.1**

Table 22: Performance on the combined set of Raganato et al. (2017), grouped by part-of-speech. Reporting F1 for WSD and P@5 for USM. MFS not applicable for USM.

9. Discussion

In this section we discuss the main findings of our work. More specifically, we discuss sense representation at specific layers of NLMs (§9.1), differences observed across models and variants (§9.2), and finally, how our sense embeddings may benefit downstream tasks (§9.3).

9.1. Layer Distribution

Throughout this article we have provided empirical evidence supporting that there is substantial non-monotonic variation in the adeptness of specific layers

of Transformer-based NLMs for sense representation. This evidence is available from both our probing analysis and the improvements in several sense-related tasks obtained from using our proposed sense profiles, which are based on non-monotonic pooling from all layers (most clearly shown in Table 3).

The cause for this variation remains elusive, calling for controlled experiments where different NLMs are tested under comparable circumstances, particularly with regards to training data and modelling objectives, although such an experimental setup may be cost-prohibitive for models of this scale. Nevertheless, seeking to better understand this variation, we conducted two qualitative experiments targeting representations of the same sentences at different layers.

In our first qualitative experiment, we compared sense similarity at different layers for the same word in context. We found some evidence potentially in support of the hypothesis advanced by Voita et al. (2019b), with the distribution of final layers resembling the distribution of the first layers moreso than the distribution of middlemost layers, where the difference between correct and incorrect senses is more marked (see example in Figure 2).

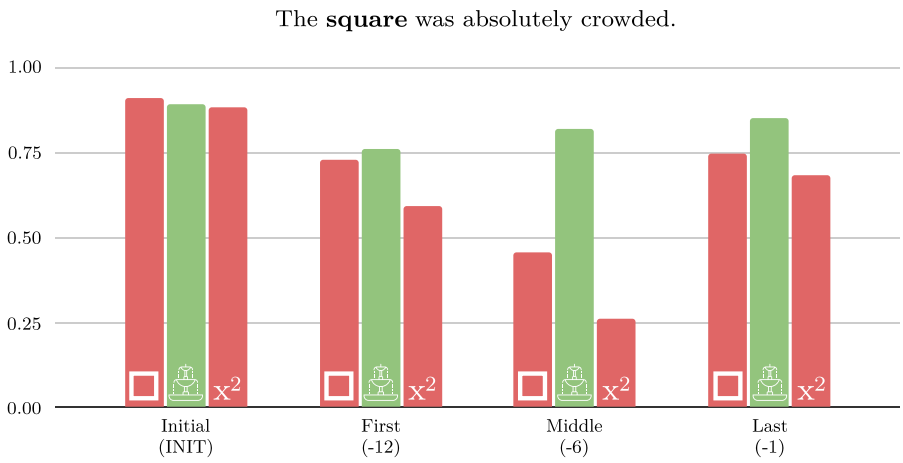


Figure 2: Cosine similarity at specific layers for the word 'square' in context and the 3 senses annotated in SemCor, represented with ALBERT-XXL. The 3 senses of square correspond to the shape, public square (correct, green bar chart) and mathematical operation, in order. Initial layer similarities are influenced by the word forms used with each sense in SemCor.

We further extended the previous experiment to a cluster-level comparison of the embedding space. For this experiment, we focus on the words present in the CoarseWSD-20 dataset (Loureiro et al., 2021), both in aggregate for measuring correct sense clustering, as well as targeting "spring" and its three distinct senses for visualization. Considering silhouette scores[12] (Rousseeuw, 1987) and PCA visualizations of the embedding space (Table 23 and Figure 3), we arrived at similar conclusions, namely that final layers tend to produce less accurate representations than layers closer to the middle, while the first layer show lowest scores. Our proposed layer pooling methods also show generally improved clustering in comparison to the sum of the last four layers. In addition, this experiment further confirms the unexpected finding regarding a different pattern of semantic representation across layers for XLNet-L, with representations from its final layer showing atypical dispersion.

We leave a more thorough large-scale analysis of this phenomenon for future work, alongside how to appropriately account for measuring the granularity of the different senses of a word, among other confounding factors.

Pooling	**BERT-L**	**XLNet-L**	**RoBERTa-L**	**ALBERT-XXL**
First Layer	0.156	0.064	0.010	0.137
Middle Layer	0.384	0.167	0.180	0.328
Final Layer	0.369	0.049	0.210	0.273
Sum Last 4	0.376	0.088	**0.218**	0.347
SP-WSD	0.377	0.250	0.203	0.387
SP-USM	**0.388**	**0.255**	0.196	**0.390**

Table 23: Mean silhouette scores for all 20 words of the 10-shot training instances (balanced) of CoarseWSD-20 (Loureiro et al., 2021). Top rows report scores for specific layers, bottom rows report scores when pooling the sum of last 4 layers and our proposed pooling strategies.

[12]We use the mean silhouette coefficients of all embeddings for a particular word to measure how well each model and pooling strategy can assign embeddings to the correct sense cluster. Silhouette coefficients are based on intra- and nearest-cluster cosine similarities. Low values represent overlapping clusters.

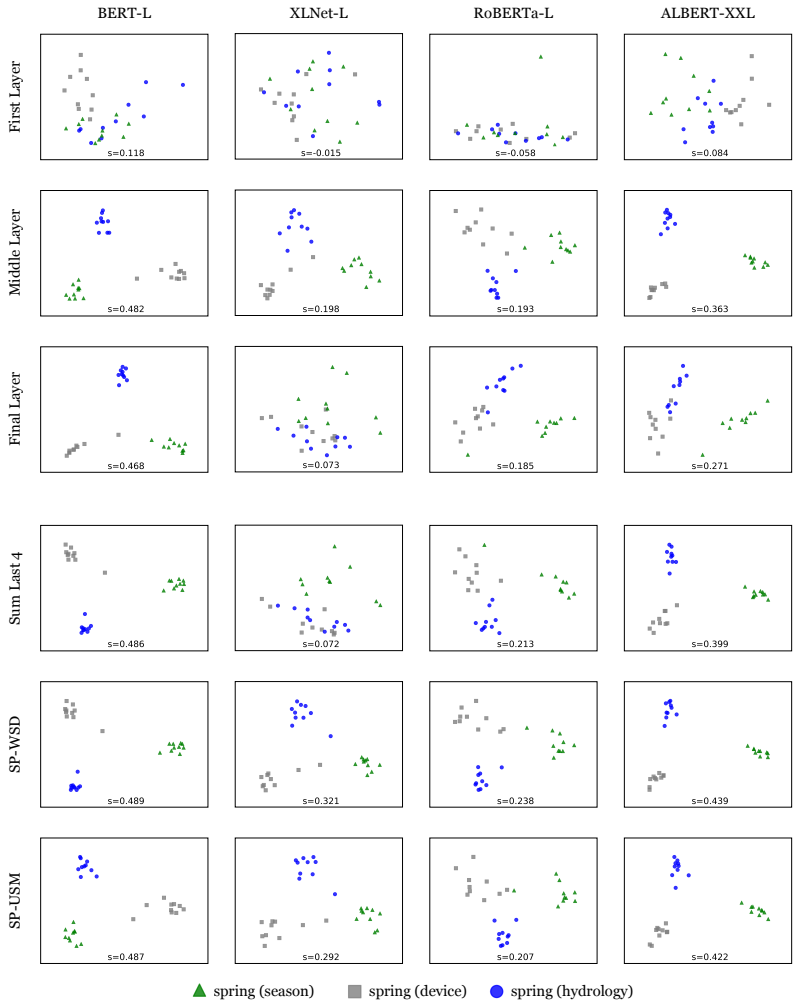


Figure 3: Visualization of embedding spaces using different pooling strategies. The last two rows correspond to our proposed pooling strategies (see §4.4 and §6). Each point corresponds to an embedding for the word "spring" in context, as provided in the 10-shot set of CoarseWSD-20 (Loureiro et al., 2021). Using PCA for dimensionality reduction. Silhouette scores s are computed before reduction.

9.2. NLM Idiosyncrasies

Besides unexpected results regarding the performance of particular layers of NLMs, we also find intriguing differences in the patterns of layer performance observed across models, and even variants of the same model. Looking at our results on Table 3, we find many intriguing examples of this variation.

For the WSD task, the most striking examples are the differences between BERT-L-UNC-WHL and any other BERT model, and the bi-modal distribution for XLNet. For example, XLNet-B exhibited its best-performing layer near the top of the model, while the best-performing layer for XLNet-L is in the bottom-half of the model. While results for USM are more consistent, we also find some peculiarities there, such as XLNet models showing worst-performing layers at the top, and ALBERT-XL showing a more biased distribution than other ALBERT variants.

The reasons for these differences in patterns across models and variants are not straightforward, specially considering many of these models are trained on very similar data and architectures. Still, among several technical differences, we highlight the differences in modelling objectives covered in Section 5.1. Out of the 4 the models we considered in this work (see performance summary on Table 24), XLNet is in fact simultaneously the model that appears most distinctive, with particularly strong performance on graded similarity tasks, and whose objectives are most different (being the only model not using MLM). Another interesting finding is that we obtain best results on WSD, USM and WiC using ALBERT-XXL, which has half the layers of the other models, but much larger embedding dimensionality (model details are available in Table 2). As for differences in variants of the same model (same objectives) we consider the possibility that trivial run-time parameters may have an impact on this variation, akin to the unexpected influence of random seeds on fine-tuning BERT models (Dodge et al., 2020).

Model	WSD (F1)	USM (P@5)	WiC (ACC)	GWCS (COR)	SID (COR)
LMMS-SP$_{\text{BERT-L}}$	75.2	86.7	67.4	76.3	77.8
LMMS-SP$_{\text{XLNet-L}}$	74.1	87.3	66.1	**78.7**	**79.5**
LMMS-SP$_{\text{RoBERTa-L}}$	75.2	86.9	67.8	75.7	74.1
LMMS-SP$_{\text{ALBERT-XXL}}$	**75.5**	**87.6**	**67.9**	75.2	77.4

Table 24: Summary comparison between different NLMs using our LMMS-SP approach.

9.3. Knowledge Integration

The ability of matching WordNet synsets to any fragment of text allows downstream applications to easily leverage the manually curated relations available on WordNet. At the same time, these sense embeddings can also serve as an entry point to many other knowledge bases linked to WordNet, such as the multilingual knowledge graph of BabelNet (Navigli & Ponzetto, 2010), the common-sense triples of ConceptNet (Speer et al., 2017) or WebChild (Tandon et al., 2017), the semantic frames of VerbNet (Schuler, 2006), and even the images of ImageNet (Russakovsky et al., 2015) or Visual Genome (Krishna et al., 2016). Several recent works have used the symbolic relations expressed in these knowledge bases to improve neural solutions to Natural Language Inference (Kapanipathi et al., 2020), Commonsense Reasoning (Lin et al., 2019), Story Generation (Ammanabrolu et al., 2020), among others.

As an example of how using LMMS-SP to bridge natural language and symbolic knowledge can be beneficial, in Figure 4 we demonstrate how these sense embeddings allow for generalization of argument spans, predicted by a semantic parser, exploiting WordNet relations between matched synsets. The matches shown in Figure 4 also illustrate how sense embeddings may be used for probing world knowledge encoded in pre-trained NLMs, as already suggested in Loureiro & Jorge (2019a).

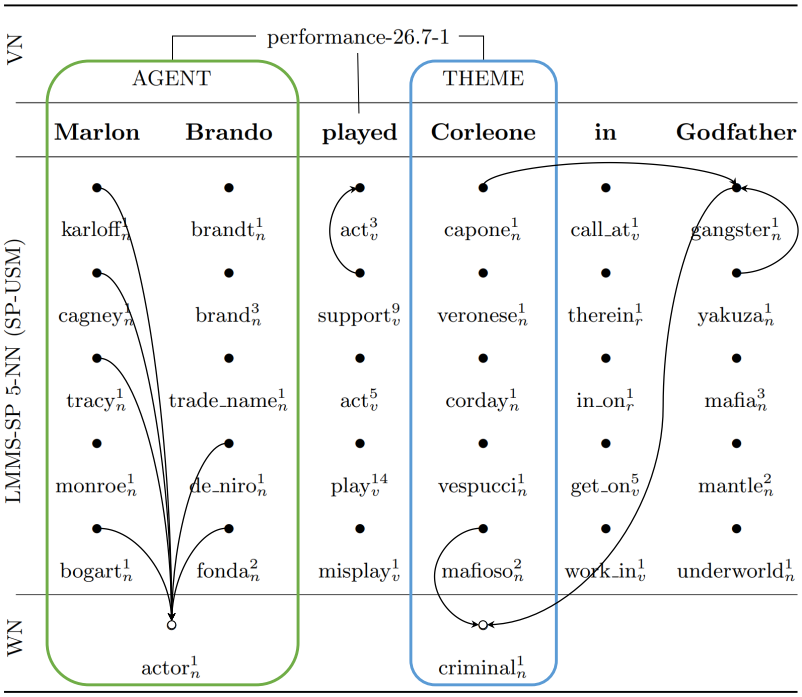


Figure 4: Example sentence with each token matched to LMMS-SP$_{\text{ALBERT-XXL}}$ sense embeddings, presenting synsets for the 5 nearest neighbors, using the SP-USM sense profile. Shows direct hypernymy relations (i.e., Is-A), included in WordNet (WN), between matched synsets, as well as hypernymy relations shared between more than one matched and unmatched synset (i.e., deducible generalizations, not in top 5 matches). Finally, at the top, we show a VerbNet (VN) semantic frame matched to this sentence, highlighting how LMMS-SP enables generalization of argument spans.

10. Conclusion

Leveraging neural language models in combination with sense-annotated corpora (and complementary resources such as glosses or relations), this work has shown that it is possible to produce sense embeddings applicable beyond mere disambiguation, with relevant implications for long-standing challenges in Artificial Intelligence such as symbol grounding.

This extension of Loureiro & Jorge (2019a) proposes a more principled

approach for learning distributional representations of word senses using pre-trained NLMs, focusing on state-of-the-art Transformer-based models. From extensive evaluation on several sense-related tasks, we demonstrated that the LMMS-SP approach is more effective than prior work at approximating precise word sense representations in the same vector space of NLMs.

The broad probing analysis of the many variants of popular NLMs endeavored in this work provides new evidence supporting further research on the interplay between pre-training objectives, layer specialization, and model size. The conclusions of this probing analysis are indeed expected to be applicable in tasks outside WSD, and for learning representations other than sense embeddings, which we leave for future work.

Effectively, there are known limitations to meaning representation based on language modelling objectives alone (Bender & Koller, 2020; Merrill et al., 2021). Nonetheless, we believe our work shows there is still much to understand about how to best leverage NLMs for meaning representation, in addition to more thoroughly testing the effectiveness of current approaches centered on self-supervision.

Release. This work is accompanied by the release of the following resources: sensekey and synset embeddings with full-coverage of WordNet based on BERT-L, XLNet-L, RoBERTa-L and ALBERT-XXL; scripts to generate embeddings following our method, using the same NLMs or others supported by the Transformers package; and scripts to run task evaluations. These resources are released under a GNU General Public License (v3) and available from this public repository: `https://github.com/danlou/lmms`

11. Acknowledgements

Daniel Loureiro is supported by the EU and Fundação para a Ciência e Tecnologia through contract DFA/BD/9028/2020 (Programa Operacional Regional Norte). Jose Camacho-Collados is supported by a UKRI Future Leaders Fellowship.

Appendices

A. Full results for SemEval 2020 - Task 3

On Table 25 we report complete leaderboard results for subtasks 1 and 2 of SemEval 2020 Task 3 (including other languages besides English), during the evaluation period.

English				**Hungarian**			
Team	**Sub1**	**Team**	**Sub2**	**Team**	**Sub1**	**Team**	**Sub2**
1. Ferryman	0.774	1. MineriaUNAM	0.723	1. N+S	0.740	1. N+S	0.658
2. will_go	0.768	**2.** $\text{LMMS}_{\text{RoBERTa-L}}$	0.720	2. Hitachi	0.681	2. Hitachi	0.616
3. MULTISEM	0.760	3. somaia	0.719	3. InfoMiner	0.754	3. MineriaUNAM	0.613
4. $\text{LMMS}_{\text{RoBERTa-L}}$	0.754	4. MULTISEM	0.718	4. Ferryman	0.774	**4.** $\text{LMMS}_{\text{XLMR-L}}$	0.565
5. InfoMiner	0.754	5. InfoMiner	0.715	**5.** $\text{LMMS}_{\text{XLMR-L}}$	0.754	5. InfoMiner	0.545

Finnish				**Slovenian**			
Team	**Sub1**	**Team**	**Sub2**	**Team**	**Sub1**	**Team**	**Sub2**
1. will_go	0.772	1. InfoMiner	0.645	1. Hitachi	0.654	1. N+S	0.579
2. Ferryman	0.745	2. N+S	0.611	2. InfoMiner	0.648	2. InfoMiner	0.573
3. N+S	0.726	3. MineriaUNAM	0.597	3. N+S	0.646	3. CitiusNLP	0.538
4. RTM	0.671	4. MULTISEM	0.492	4. CitiusNLP	0.624	4. tthhanh	0.516
11. $\text{LMMS}_{\text{XLMR-L}}$	0.360	**7.** $\text{LMMS}_{\text{XLMR-L}}$	0.354	**8.** $\text{LMMS}_{\text{XLMR-L}}$	0.560	**9.** $\text{LMMS}_{\text{XLMR-L}}$	0.483

Table 25: Results from the leaderboard of subtasks 1 and 2 of SemEval 2020 Task 3 - Predicting the (Graded) Effect of Context in Word Similarity. Rank reported in team names. At the time of this evaluation, we did not use the sense profiles proposed in this paper, so our reported results on this table are based on senses embeddings pooled from the last 4 layers of the specified models, following Loureiro & Jorge (2019a).

B. Stanford Contextual Word Similarities (SCWS)

On Table 26 we report our results on the Stanford Contextual Word Similarities (Huang et al., 2012, SCWS) task. We address this task similarly to GWCS (see Section 7.4). Given two words in context, each within an independent sentence, we disambiguate both occurrences and score each pair as the average of similarities between corresponding sense and contextual embeddings.

Results on SCWS follow performance on GWCS, with XLNet-L outperforming other NLMs as well as results from related works. Analysing performance by Part-of-Speech (POS), we find that nouns appear most challenging for this task, particularly when being compared against other nouns.

System	**ALL** (n=2003)	**N-N** (n=1328)	**N-V** (n=140)	**N-A** (n=30)	**V-V** (n=399)	**V-A** (n=9)	**A-A** (n=97)
Huang et al. (2012)	65.7	–	–	–	–	–	–
SensEmbed (2015)	62.4	–	–	–	–	–	–
NASARI (2016)	–	47.1	–	–	–	–	–
DeConf (2016)	71.5	–	–	–	–	–	–
LessLex (2020a)	69.5	69.2	69.6	82.0	64.1	73.6	63.8
ARES (2020b)	67.9	66.6	68.6	**87.9**	67.2	66.7	69.4
BERT-L (SP-WSD)	59.3	56.8	67.4	78.4	59.4	60.0	61.1
XLNet-L (SP-WSD)	73.9	71.6	75.6	81.3	75.8	**78.3**	76.0
RoBERTa-L (SP-WSD)	63.8	59.1	71.3	66.6	68.7	73.3	66.7
ALBERT-XXL (SP-WSD)	65.9	63.7	69.4	74.9	66.3	75.0	69.5
LMMS-SP$_{\text{BERT-L}}$	64.1	62.3	67.1	82.6	63.5	51.7	68.3
LMMS-SP$_{\text{XLNet-L}}$	**75.9**	**73.7**	**75.8**	81.5	**78.0**	75.0	**79.4**
LMMS-SP$_{\text{RoBERTa-L}}$	67.4	63.4	73.9	70.8	71.1	75.0	68.1
LMMS-SP$_{\text{ALBERT-XXL}}$	69.9	68.8	72.4	76.4	69.9	66.6	70.9

Table 26: Results on SCWS (Spearman correlation scores, $\rho \times 100$), considering the entire set of pairs (ALL) as well as results for subsets pairing particular Parts-of-Speech (with n denoting the number of instances for each subset), similarly to Colla et al. (2020a).

References

Ammanabrolu, P., Tien, E., Cheung, W., Luo, Z., Ma, W., Martin, L. J., & Riedl, M. O. (2020). Story realization: Expanding plot events into sentences. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*, 7375–7382. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6232. doi:10.1609/aaai.v34i05.6232.

Armendariz, C. S., Purver, M., Pollak, S., Ljubešić, N., Ulčar, M., Vulić, I., & Pilehvar, M. T. (2020a). SemEval-2020 task 3: Graded word similarity in context. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 36–49). Barcelona (online): International Committee for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.semeval-1.3.

Armendariz, C. S., Purver, M., Ulčar, M., Pollak, S., Ljubešić, N., & Granroth-Wilding, M. (2020b). CoSimLex: A resource for evaluating graded word similarity in context. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 5878–5886). Marseille, France: European Language Resources Association. URL: https://www.aclweb.org/anthology/2020.lrec-1.720.

Athiwaratkun, B., Wilson, A., & Anandkumar, A. (2018). Probabilistic FastText for multi-sense word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1–11). Melbourne, Australia: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P18-1001. doi:10.18653/v1/P18-1001.

Barba, E., Procopio, L., & Navigli, R. (2021). ConSeC: Word sense disambiguation as continuous sense comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 1492–1503). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. URL: https://aclanthology.org/2021.emnlp-main.112.

Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 5185–5198). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.463. doi:10.18653/v1/2020.acl-main.463.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003a). A neural probabilistic language model. *J. Mach. Learn. Res.*, *3*, 1137–1155.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003b). A neural probabilistic language model. *Journal of machine learning research*, *3*, 1137–1155.

Bevilacqua, M., & Navigli, R. (2020). Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 2854–2864). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.255. doi:10.18653/v1/2020.acl-main.255.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*, 993–1022.

Blevins, T., & Zettlemoyer, L. (2020). Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1006–1017). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.95. doi:10.18653/v1/2020.acl-main.95.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. URL: https://www.aclweb.org/anthology/Q17-1010. doi:10.1162/tacl_a_00051.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. *CoRR*, *abs/2005.14165*. URL: https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

Cai, Z. G., Gilbert, R. A., Davis, M. H., Gaskell, M. G., Farrar, L., Adler, S., & Rodd, J. M. (2017). Accent modulates access to word meaning: Evidence for a speaker-model account of spoken word recognition. *Cognitive Psychology*, *98*, 73 – 101. URL: http://www.sciencedirect.com/science/article/pii/S0010028517300762. doi:https://doi.org/10.1016/j.cogpsych.2017.08.003.

Camacho-Collados, J., & Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Int. Res.*, *63*, 743–788. URL: https://doi.org/10.1613/jair.1.11259. doi:10.1613/jair.1.11259.

Camacho-Collados, J., Pilehvar, M. T., Collier, N., & Navigli, R. (2017). SemEval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 15–26). Vancouver, Canada: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/S17-2002. doi:10.18653/v1/S17-2002.

Camacho-Collados, J., Pilehvar, M. T., & Navigli, R. (2015). NASARI: a novel approach to a semantically-aware representation of items. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 567–577). Denver, Colorado: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N15-1059. doi:10.3115/v1/N15-1059.

Camacho-Collados, J., Pilehvar, M. T., & Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, *240*, 36 – 64. URL: http://www.sciencedirect.com/science/article/pii/S0004370216300820. doi:https://doi.org/10.1016/j.artint.2016.07.005.

Chronis, G., & Erk, K. (2020). When is a bishop not like a rook? when it's like a rabbi! multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning* (pp. 227–244). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.conll-1.17. doi:10.18653/v1/2020.conll-1.17.

Colla, D., Mensa, E., & Radicioni, D. P. (2020a). LessLex: Linking multilingual embeddings to SenSe representations of LEXical items. *Computational Linguistics*, *46*, 289–333. URL: https://www.aclweb.org/anthology/2020.cl-2.3. doi:10.1162/coli_a_00375.

Colla, D., Mensa, E., & Radicioni, D. P. (2020b). Novel metrics for computing semantic similarity with sense embeddings. *Knowledge-Based Systems*, *206*, 106346. URL: https://www.sciencedirect.com/science/article/pii/S0950705120305025. doi:https://doi.org/10.1016/j.knosys.2020.106346.

Colla, D., Mensa, E., & Radicioni, D. P. (2020c). Sense identification data: A dataset for lexical semantics. *Data in Brief*, *32*, 106267. URL: https://www.sciencedirect.com/science/article/pii/S2352340920311616. doi:https://doi.org/10.1016/j.dib.2020.106267.

Collobert, R., & Weston, J. (2007). Fast semantic extraction using a novel neural network architecture. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (pp. 560–567). Prague, Czech

Republic: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P07-1071.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160–167).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, *12*, 2493–2537.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., & Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2978–2988). Florence, Italy: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P19-1285. doi:10.18653/v1/P19-1285.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*, 391–407.

Deerwester, S. C., Dumais, S. T., Furnas, G. W., Harshman, R. A., Landauer, T. K., Lochbaum, K. E., & Streeter, L. A. (1989). Computer information retrieval using latent semantic structure. US Patent 4,839,853.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N19-1423. doi:10.18653/v1/N19-1423.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. arXiv:2002.06305.

Dong, Z., Dong, Q., & Hao, C. (2006). Hownet and the computation of meaning.

Erk, K. (2016). What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, *9*, 1–63. doi:10.3765/sp.9.17.

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 55–65). Hong Kong, China: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D19-1006. doi:10.18653/v1/D19-1006.

Fellbaum, C. (1998). Wordnet : an electronic lexical database. MIT Press.

Firth, J. (1957). A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford. Reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

Firth, J. R. (1935). The technique of semantics. *Transactions of the Philological Society*, *34*, 36–73.

Flekova, L., & Gurevych, I. (2016). Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2029–2041). Berlin, Germany: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P16-1191. doi:10.18653/v1/P16-1191.

Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, *10*, 1–309.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In D. Precup, & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning* (pp. 1321–1330). PMLR volume 70 of *Proceedings of Machine Learning Research*. URL: http://proceedings.mlr.press/v70/guo17a.html.

Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1489–1501). Berlin, Germany: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P16-1141. doi:10.18653/v1/P16-1141.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*, 146–162.

Hewitt, J., & Manning, C. D. (2019). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4129–4138). Minneapolis, Minnesota: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N19-1419. doi:10.18653/v1/N19-1419.

Huang, E., Socher, R., Manning, C., & Ng, A. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 873–882). Jeju Island, Korea: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P12-1092.

Huang, L., Sun, C., Qiu, X., & Huang, X. (2019). GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th*

International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3509–3514). Hong Kong, China: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D19-1355. doi:10.18653/v1/D19-1355.

Iacobacci, I., Pilehvar, M. T., & Navigli, R. (2015). SensEmbed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 95–105). Beijing, China: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P15-1010. doi:10.3115/v1/P15-1010.

Ide, N., Baker, C. F., Fellbaum, C., & Passonneau, R. J. (2010). The manually annotated sub-corpus: A community resource for and by the people. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (Short Papers)* (pp. 68–73). Uppsala, Sweden.

Kapanipathi, P., Thost, V., Sankalp Patel, S., Whitehead, S., Abdelaziz, I., Balakrishnan, A., Chang, M., Fadnis, K., Gunasekara, C., Makni, B., Mattei, N., Talamadupula, K., & Fokoue, A. (2020). Infusing knowledge into the textual entailment task using graph convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*, 8074–8081. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6318. doi:10.1609/aaai.v34i05.6318.

Klein, D. E., & Murphy, G. L. (2001). The representation of polysemous words. *Journal of Memory and Language*, *45*, 259 – 282. URL: http://www.sciencedirect.com/science/article/pii/S0749596X01927792. doi:https://doi.org/10.1006/jmla.2001.2779.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M., & Fei-Fei, L. (2016).

In *Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations*. URL: https://arxiv.org/abs/1602.07332.

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 66–71). Brussels, Belgium: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D18-2012. doi:10.18653/v1/D18-2012.

Kuznetsov, I., & Gurevych, I. (2020). A matter of framing: The impact of linguistic formalism on probing results. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 171–182). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.emnlp-main.13. doi:10.18653/v1/2020.emnlp-main.13.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H1eA7AEtvS.

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*, 211.

Levine, Y., Lenz, B., Dagan, O., Ram, O., Padnos, D., Sharir, O., Shalev-Shwartz, S., Shashua, A., & Shoham, Y. (2020). SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 4656–4667). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.423. doi:10.18653/v1/2020.acl-main.423.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence

pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7871–7880). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.

Li, J., & Jurafsky, D. (2015). Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1722–1732). Lisbon, Portugal: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D15-1200. doi:10.18653/v1/D15-1200.

Lin, B. Y., Chen, X., Chen, J., & Ren, X. (2019). KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 2829–2839). Hong Kong, China: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D19-1282. doi:10.18653/v1/D19-1282.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., & Smith, N. A. (2019a). Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 1073–1094). Minneapolis, Minnesota: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N19-1112. doi:10.18653/v1/N19-1112.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019b). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, *abs/1907.11692*. URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.

Loureiro, D., & Camacho-Collados, J. (2020). Don't neglect the obvious: On

the role of unambiguous words in word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 3514–3520). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.emnlp-main.283. doi:10.18653/v1/2020.emnlp-main.283.

Loureiro, D., & Jorge, A. (2019a). Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5682–5691). Florence, Italy: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P19-1569. doi:10.18653/v1/P19-1569.

Loureiro, D., & Jorge, A. (2019b). LIAAD at SemDeep-5 challenge: Word-in-context (WiC). In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)* (pp. 1–5). Macau, China: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W19-5801.

Loureiro, D., Rezaee, K., Pilehvar, M. T., & Camacho-Collados, J. (2021). Analysis and Evaluation of Language Models for Word Sense Disambiguation. *Computational Linguistics*, (pp. 1–55). URL: https://doi.org/10.1162/coli_a_00405. doi:10.1162/coli_a_00405.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, *28*, 203–208.

Mancini, M., Camacho-Collados, J., Iacobacci, I., & Navigli, R. (2017). Embedding words and senses together via joint knowledge-enhanced training. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)* (pp. 100–111). Vancouver, Canada: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/K17-1012. doi:10.18653/v1/K17-1012.

McCoy, T., Pavlick, E., & Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3428–3448). Florence, Italy: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P19-1334. doi:10.18653/v1/P19-1334.

Mcdonald, S., & Ramscar, M. (2001). Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *In Proceedings of the 23rd Annual Conference of the Cognitive Science Society* (pp. 611–6).

Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (pp. 51–61). Berlin, Germany: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/K16-1006. doi:10.18653/v1/K16-1006.

Merrill, W., Goldberg, Y., Schwartz, R., & Smith, N. A. (2021). Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? arXiv:2104.10809.

Meyer, C. M., & Gurevych, I. (2012). Wiktionary: a new rival for expert-built lexicons? exploring the possibilities of collaborative lexicography.

Mickus, T., Paperno, D., Constant, M., & van Deemter, K. (2020). What do you mean, bert? assessing bert as a distributional semantics model. *Proceedings of the Society for Computation in Linguistics*, *3*. doi:10.7275/t778-ja71.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* NIPS'13 (p. 3111–3119). Red Hook, NY, USA: Curran Associates Inc.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miller, G. A., Chodorow, M., Landes, S., Leacock, C., & Thomas, R. G. (1994). Using a semantic concordance for sense identification. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*. URL: https://www.aclweb.org/anthology/H94-1046.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys*, *41*, 10:1–10:69. URL: http://doi.acm.org/10.1145/1459352.1459355. doi:10.1145/1459352.1459355.

Navigli, R., & Ponzetto, S. P. (2010). In *BabelNet: Building a Very Large Multilingual Semantic Network* (pp. 216–225).

Neelakantan, A., Shankar, J., Passos, A., & McCallum, A. (2014). Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1059–1069). Doha, Qatar: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D14-1113. doi:10.3115/v1/D14-1113.

Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). *The measurement of meaning*. 47. University of Illinois press.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)* (pp. 48–53). Minneapolis, Minnesota: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N19-4009. doi:10.18653/v1/N19-4009.

Pasini, T. (2020). The knowledge acquisition bottleneck problem in multilingual word sense disambiguation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-20, Yokohama, Japan*.

Pelevina, M., Arefiev, N., Biemann, C., & Panchenko, A. (2016). Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP* (pp. 174–183). Berlin, Germany: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W16-1620. doi:10.18653/v1/W16-1620.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D14-1162. doi:10.3115/v1/D14-1162.

Pereira, F., Lou, B., Pritchett, B., Ritter, S., Gershman, S. J., Kanwisher, N., Botvinick, M., & Fedorenko, E. (2018). Toward a universal decoder of linguistic meaning from brain activation. *Nature communications*, *9*, 1–13.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018a). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 2227–2237). New Orleans, Louisiana: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N18-1202. doi:10.18653/v1/N18-1202.

Peters, M., Neumann, M., Zettlemoyer, L., & Yih, W.-t. (2018b). Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 1499–1509). Brussels, Belgium: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D18-1179. doi:10.18653/v1/D18-1179.

Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 43–54). Hong Kong, China: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D19-1005. doi:10.18653/v1/D19-1005.

Piantadosi, S. T., Tily, H., & Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition*, *122*, 280 – 291. URL: http://www.sciencedirect.com/science/article/pii/S0010027711002496. doi:https://doi.org/10.1016/j.cognition.2011.10.004.

Pilehvar, M. T., & Camacho-Collados, J. (2019). WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 1267–1273). Minneapolis, Minnesota: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N19-1128. doi:10.18653/v1/N19-1128.

Pilehvar, M. T., Camacho-Collados, J., Navigli, R., & Collier, N. (2017). Towards a seamless integration of word senses into downstream NLP applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1857–1869). Vancouver, Canada: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P17-1170. doi:10.18653/v1/P17-1170.

Pilehvar, M. T., & Collier, N. (2016). De-conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1680–1690). Austin, Texas: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D16-1174. doi:10.18653/v1/D16-1174.

Radach, R., Deubel, H., Vorstius, C., & Hofmann (eds.), M. (2017). Abstracts of the 19th european conference on eye movements 2017. *Journal of Eye Movement Research*, *10*. URL: https://bop.unibe.ch/JEMR/article/view/JEMR.10.6.1. doi:10.16910/jemr.10.6.1.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners, .

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*, 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

Raganato, A., Camacho-Collados, J., & Navigli, R. (2017). Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (pp. 99–110). Valencia, Spain: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/E17-1010.

Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Coenen, A., Pearce, A., & Kim, B. (2019). Visualizing and measuring the geometry of bert. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlche Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (pp. 8594–8603). Curran Associates, Inc. volume 32. URL: https://proceedings.neurips.cc/paper/2019/file/159c1ffe5b61b41b3c4d8f4c2150f6c4-Paper.pdf.

Reisinger, J., & Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 109–117). Los Angeles, California: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N10-1013.

Rodd, J. M. (2020). Settling into semantic space: An ambiguity-focused account

of word-meaning access. *Perspectives on Psychological Science*, *15*, 411–427. URL: https://doi.org/10.1177/1745691619885860. doi:10.1177/1745691619885860. arXiv:https://doi.org/10.1177/1745691619885860. PMID: 31961780.

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866. URL: https://www.aclweb.org/anthology/2020.tacl-1.54. doi:10.1162/tacl_a_00349.

Rothe, S., & Schütze, H. (2015). AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1793–1803). Beijing, China: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P15-1173. doi:10.3115/v1/P15-1173.

Rousseeuw, P. (1987). Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. comput. appl. math. 20, 53-65. *Journal of Computational and Applied Mathematics*, *20*, 53–65. doi:10.1016/0377-0427(87)90125-7.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115*, 211–252. doi:10.1007/s11263-015-0816-y.

Salton, G. (1971). The smart system. *Retrieval Results and Future Plans*, .

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*, 613–620.

Scarlini, B., Pasini, T., & Navigli, R. (2020a). SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation.

In *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence* (pp. 8758–8765). Association for the Advancement of Artificial Intelligence.

Scarlini, B., Pasini, T., & Navigli, R. (2020b). With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 3528–3539). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.emnlp-main.285. doi:10.18653/v1/2020.emnlp-main.285.

Schuler, K. K. (2006). *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis University of Pennsylvania. URL: http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf.

Schutze, H. (1992). Dimensions of meaning. In *Supercomputing'92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing* (pp. 787–796). IEEE.

Soler, A. G., & Apidianaki, M. (2021). Let's play mono-poly: Bert can reveal words' polysemy level and partitionability into senses. *Transactions of the Association for Computational Linguistics (TACL)*, .

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* AAAI'17 (p. 4444–4451). AAAI Press.

Tandon, N., de Melo, G., & Weikum, G. (2017). WebChild 2.0 : Fine-grained commonsense knowledge distillation. In *Proceedings of ACL 2017, System Demonstrations* (pp. 115–120). Vancouver, Canada: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P17-4020.

Tenney, I., Das, D., & Pavlick, E. (2019a). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4593–4601). Florence, Italy: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P19-1452. doi:10.18653/v1/P19-1452.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S. R., Das, D., & Pavlick, E. (2019b). What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SJzSgnRcKX.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vial, L., Lecouteux, B., & Schwab, D. (2018). UFSAC: Unification of sense annotated corpora and tools. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L18-1166.

Vial, L., Lecouteux, B., & Schwab, D. (2019). Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation. In *Proceedings of the 10th Global Wordnet Conference* (pp. 108–117). Wroclaw, Poland: Global Wordnet Association. URL: https://www.aclweb.org/anthology/2019.gwc-1.14.

Voita, E., Sennrich, R., & Titov, I. (2019a). The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 4396–4406). Hong Kong, China: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D19-1448. doi:10.18653/v1/D19-1448.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., & Titov, I. (2019b). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5797–5808). Florence, Italy: Association

for Computational Linguistics. URL: https://www.aclweb.org/anthology/P19-1580. doi:10.18653/v1/P19-1580.

Vu, T., & Parker, D. S. (2016). k-embeddings: Learning conceptual embeddings for words using context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1262–1267). San Diego, California: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/N16-1151. doi:10.18653/v1/N16-1151.

Vulić, I., Ponti, E. M., Litschko, R., Glavaš, G., & Korhonen, A. (2020). Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 7222–7240). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.emnlp-main.586. doi:10.18653/v1/2020.emnlp-main.586.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlche Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. volume 32. URL: https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Wittgenstein, L. (1953). Philosophical investigations, trans. *GEM Anscombe*, *261*, 49.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demon-*

strations (pp. 38–45). Online: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

Yaghoobzadeh, Y., & Schütze, H. (2016). Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 236–246). Berlin, Germany: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P16-1023. doi:10.18653/v1/P16-1023.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5753–5763).

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics* (pp. 189–196). Cambridge, Massachusetts, USA: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P95-1026. doi:10.3115/981658.981684.

Yuan, D., Richardson, J., Doherty, R., Evans, C., & Altendorf, E. (2016). Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 1374–1385). Osaka, Japan: The COLING 2016 Organizing Committee. URL: https://www.aclweb.org/anthology/C16-1130.

Zhou, X., Sap, M., Swayamdipta, S., Choi, Y., & Smith, N. (2021). Challenges in automated debiasing for toxic language detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 3143–3155). Online: Association

for Computational Linguistics. URL: https://www.aclweb.org/anthology/2021.eacl-main.274.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision* (pp. 19–27).

Appendix H

Precisely Probing Commonsense Knowledge in Pretrained Language Models using Sense Embeddings

Daniel Loureiro and Alípio Jorge. 2022. Under Review.

Relevant Contributions

- Introduces SynBERT, a BERT model augmented with sense embeddings for the full WordNet - allows for precise use of synsets with NLMs, as if they are regular tokens.
- Presents SenseLAMA, a challenging commonsense probing test set that targets WordNet senses and reveals which commonsense relations appear more tractable for NLMs.

Return to Table of Contents

Precisely Probing Commonsense Knowledge in Pretrained Language Models using Sense Embeddings

Daniel Loureiro, Alípio Mário Jorge
LIAAD - INESC TEC
Faculty of Sciences - University of Porto, Portugal
daniel.b.loureiro@inesctec.pt, amjorge@fc.up.pt

Abstract

Progress on commonsense reasoning is usually measured from performance improvements on Question Answering tasks specifically designed to require commonsense knowledge. However, fine-tuning large Language Models (LMs) on these specific tasks does not directly evaluate commonsense learned during pretraining. The most direct assessments of commonsense knowledge in pretrained LMs are arguably cloze-style prompting tasks targeting commonsense assertions (e.g., A pen is used for `[MASK]`.). However, this approach is restricted by the LM's vocabulary available for masked predictions, and its precision is subject to the context provided by the assertion.

In this work, we present a method for enriching LMs with a grounded sense inventory (i.e., WordNet) available at the vocabulary level without further training. This modification augments the prediction space of cloze-style prompts to the size of a large ontology while enabling finer-grained (sense-level) queries and predictions. In order to evaluate LMs with higher precision, we propose SenseLAMA, a cloze-style task featuring verbalized relations from disambiguated triples sourced from WordNet, WikiData, and ConceptNet. Applying our method to BERT, producing a WordNet-enriched version named SynBERT, we find that LMs can learn non-trivial commonsense knowledge from self-supervision, covering numerous relations, and more effectively than comparable similarity-based approaches.

1 Introduction

A relatively new direction for benchmarking Language Models (LMs) are tasks designed to require commonsense knowledge and reasoning. These tasks usually target commonsense concepts under a Question Answering (QA) format (Mihaylov et al., 2018; Talmor et al., 2019; Bisk et al., 2020; Nie et al., 2020) and follow scaling trends. Increasing the model's parameters leads to improved results, specially in few-shot learning settings (Chowdhery et al., 2022). Hybrid methods, particularly those fusing LMs with Graph Neural Networks, have shown that Commonsense Knowledge Graphs (CKGs) can help improve performance on these tasks (Xu et al., 2021; Yasunaga et al., 2021; Zhang et al., 2022). The results obtained by these works, using relatively small LMs, suggest that CKGs can be an alternative (or complement) to increasing model size, with the added benefit of supporting more interpretable results.

Nevertheless, the QA approach provides only an indirect measure of a pretrained model's ability to understand and reason with commonsense concepts. The models attaining best results on these tasks are often too large for thorough analysis, and the QA format can promote shallow learning from annotation artifacts or spurious cues unrelated to commonsense (Branco et al., 2021).

There are more direct ways of evaluating commonsense knowledge in LMs, such as scoring generated triples (Davison et al., 2019), infilling cloze-style statements (Petroni et al., 2019), or fine-tuning for explicit generation of commonsense statements (Bosselut et al., 2019). However, these approaches are either limited by each LM's particular vocabulary or biased by the available training data (Wang et al., 2021). Additionally, existing tasks and methods do not target grounded representations, which is essential for high-precision CKGs (Tandon et al., 2014; Dalvi Mishra et al., 2017), and context-independent reference (Eyal et al., 2022).

Commonsense tasks and approaches typically leverage ConceptNet (Speer et al., 2017), a popular CKG built from an extensive crowdsourcing effort (Storks et al., 2019). Although ConceptNet is arguably the most popular CKG available, its nodes are composed of free-form text rather than disambiguated (canonical) representations, allowing for misleading associations and aggravating the network's sparsity (Li et al., 2016; Jastrzęb-

ski et al., 2018; Wang et al., 2020). The WordNet (Miller, 1992) sense inventory is a natural choice for a set of ontologically grounded concept-level representations, having been curated by experts over decades and spanning various knowledge domains and syntactic categories of the English language. Recent developments on WSD and Uninformed Sense Matching (USM) have shown that WordNet senses can be mapped to naturally occurring sentences with high precision (Loureiro et al., 2022), including at higher-abstraction levels (e.g., 'Marlon Brando' to actor^1_n). WordNet's utility for commonsense tasks is limited by its narrow set of relations, focused on lexical relations (mostly hypernymy). However, its smaller size, compared to WikiData (Vrandečić and Krötzsch, 2014) or BabelNet (Navigli and Ponzetto, 2012), for example, also presents an opportunity for effective expansion with reduced sparsity, which is important for symbolic reasoning (Huang et al., 2021).

In this work, we propose that a LM augmented with explicit sense-level representations (see Figure 1) may present a solution for precise evaluation of commonsense knowledge learned during pretraining that is not limited by the LM's vocabulary. Additionally, we explore how this enriched model can be used for grounded commonsense relation extraction towards precise and unbiased (w.r.t. commonsense training data) CKG construction that hybrid approaches may use. Considering there is currently no set of grounded assertions available to assess progress in this direction, we propose a cloze-style probing task targeting specific senses and commonsense relations, inspired by Petroni et al. (2019). Our contributions[1] are the following:

- A BERT[2] model with 117k new sense-specific embeddings added to its vocabulary, based on the model's own internal states (SynBERT).
- The SenseLAMA probing task targeting wide-ranging and precise commonsense – based on WordNet, WikiData, and ConceptNet.
- Analyses on the impact of different input types for eliciting accurate commonsense knowledge from BERT.
- A new CKG grounded on WordNet with 23k unseen triples over 18 commonsense relations (e.g., *UsedFor*) generated by prompting.

[1] https://github.com/anonymous/synbert

[2] While we focus on BERT and WordNet, our methods are broadly applicable to LMs and alternative representations.

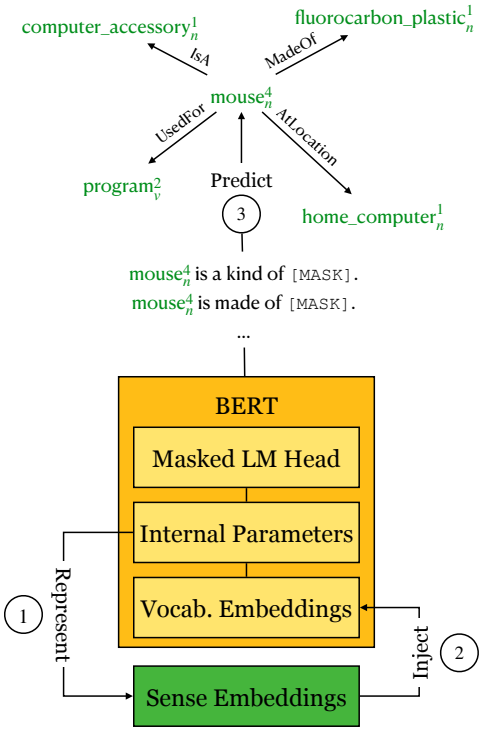


Figure 1: Our 3-step method for extracting unsupervised commonsense relations between concepts (i.e., word senses) from pretrained language models. Relations are expressed as verbalizations that may be exchanged to target any other property of interest.

2 Related Work

Large LMs have featured prominently in the latest efforts to build richer and more accurate CKGs. COMET (Hwang et al., 2021) is a generative model based on BART (Lewis et al., 2020) trained on ConceptNet and ATOMIC (Sap et al., 2019) and proven capable of producing novel accurate triples for challenging relation types, such as *HinderedBy*. More recently, West et al. (2021) have proposed ATOMIC-10x, which leverages generated text from GPT-3 (Brown et al., 2020) in combination with a critic model to create the largest and most accurate semi-automatically constructed CKG. This accuracy was determined using both qualitative human ratings and quantitative measures. However, these works are primarily concerned with extracting large CKGs using fine-tuned or distilled LMs, and do not focus on directly evaluating the CSK learned during pretraining. Additionally, these works do not target grounded representations, considering only relations between free-text nodes, similarly to ConceptNet.

Solving for both disambiguated representations and sparsity resulting from free-text redundancy, WebChild (Tandon et al., 2014) proposes a CKG, grounded on WordNet senses, assembled from label propagation and pattern matching on Web corpora. WebChild features a large CKG (over 4M triples), but it predates large contextual LMs and the ensuing progress in WSD, making this resource unreliable by current standards. Recent works on CKGs also focus on other aspects besides size and accuracy, such as salience (Chalier et al., 2020) or alternatives to triples (Nguyen et al., 2021).

Our work is most related to LAMA (Petroni et al., 2019), which compiles masked assertions based on triples from ConceptNet and other resources, and measures how many triples can be accurately recovered when masking the object term. However, LAMA was designed for single-token masked prediction based on the intersection of the subword or byte-level token vocabularies used by the particular set of LMs considered in that work[3]. Consequently, LAMA is limited by design to a total of 21k prediction candidates.

LAMA is an important early result of LM probing, but besides the previously mentioned technical limitations, its findings have also been challenged in later works. Kassner and Schütze (2020) demonstrated that LMs are susceptible to mispriming and often unable to handle negation. Poerner et al. (2020) further showed that LMs could be biased by the surface form of entity names. Moreover, Dufter et al. (2021) found that static embeddings using a nearest neighbors (kNN) approach can outperform LMs on the LAMA benchmark, casting doubt on the presumed advantages of LMs for the task. Still, LAMA inspired others to use knowledge graphs (KGs) generated by LMs for intrinsic evaluation. Swamy et al. (2021) proposes extracting KGs from LMs to support interpretability and direct comparison between different LMs, or training stages. Aspillaga et al. (2021) follows a similar direction but proposes evaluating extracted KGs by concept relatedness using hypernymy relations from WordNet and sense-tagged glosses.

Our approach overcomes the vocabulary limitations of LAMA while outperforming a comparable kNN baseline. We also explore using extracted CKGs to evaluate LMs, alongside the generation of novel CKGs.

[3]This limitation stems from the fact that each word may be split into several tokens, whose number conditions predictions to words that match it and is specific to each LM's tokenizer.

3 SenseLAMA

We begin by describing our probing task to evaluate the commonsense knowledge learned during LM pretraining. SenseLAMA features verbalized relations[4] between word senses from triples sourced from WordNet, WikiData, and ConceptNet. In the following, we describe how we compiled SenseLAMA using these resources, including mapping triples to specific WordNet senses (i.e., synsets).

Unlike other works (e.g., Feng et al., 2020), we do not merge similar relations. Since our approach is unsupervised, we do not benefit from additional examples per relation. Thus, we prefer preserving performance metrics specific to each source.

We use the core WordNet synsets, initially defined by Boyd-Graber et al. (2005), to create an easier subset of SenseLAMA. While the full WordNet covers over 117k synsets, core synsets are restricted to the 5k[5] most frequently occurring word senses, dramatically reducing the number of prediction candidates. Thus, our 'Core' subset is derived from the 'Full' SenseLAMA, including only instances where both arguments of the triple belong to the set of core WordNet synsets. If this filter results in a relation with less than ten instances, that relation is discarded from the 'Core' subset. Table 1 reports counts for each source and relation in SenseLAMA.

WordNet Our base ontology already contains several relations which arguably fall under the scope of commonsense knowledge, such as hypernymy, meronymy, or antonymy. Since these relations already target synsets within WordNet, no additional mapping or disambiguation is required. Very frequent relations are capped at 10k samples.

WikiData This vast resource contains millions of triples for thousands of relations. We only consider a few select relations most associated with commonsense knowledge. Furthermore, we only admit triples for which the head and tail can be mapped to WordNet v3.0, either via the direct link available in WikiData's item properties or through linking to BabelNet, which we map to WordNet using the mapping from Navigli and Ponzetto (2012). Alternatively, we map some triples via hapax linking (McCrae and Cillessen, 2021), when the triple's arguments correspond to unambiguous words.

[4]Appendix A shows handcrafted templates used for WordNet and WikiData triples, following Petroni et al. (2019).

[5]Only 4,960 synsets can be mapped to WordNet v3.0.

ConceptNet We focus on the OMCS subset of ConceptNet, which includes full sentences collected through crowdsourcing, together with the corresponding triples. Using these sentences, we do not require templates and can provide systems with the same input presented to crowd workers. We align arguments within sentences using the KMP algorithm (Knuth et al., 1977) and disambiguate those words in context using ESC (Barba et al., 2021), a state-of-the-art WSD system. Triples that cannot be successfully aligned are discarded. For added precision, we constrain WSD according to each relation's particular Part-of-Speech types (Havasi et al., 2009).

Source/Relation	Core	Full
WordNet (WN)	1,757	41,237
Hypernym	1,288	10,000
Holonym (Member)	26	10,000
Holonym (Part)	145	7,832
Antonym	282	7,391
Hypernym (Instance)	-	5,356
Meronym (Substance)	16	658
WikiData (WD)	136	7,222
P31 (Instance of)	39	2,968
P361 (Part of)	45	1,367
P366 (Use)	27	763
P186 (Made from)	15	639
P461 (Opposite of)	10	501
P737 (Influenced by)	-	316
P2283 (Uses)	-	268
P463 (Member of)	-	183
P1535 (Used by)	-	151
P279 (Subclass of)	-	66
ConceptNet (CN)	1,032	3,541
AtLocation	328	1,004
UsedFor	162	661
IsA	120	512
Causes	38	224
HasSubevent	57	205
HasPrerequisite	50	165
HasProperty	47	157
CapableOf	48	123
MotivatedByGoal	37	104
HasA	48	97
PartOf	33	80
CausesDesire	14	52
ReceivesAction	19	44
MadeOf	18	42
Desires	13	28
CreatedBy	-	17
HasFirstSubevent	-	14
HasLastSubevent	-	12
All	2,925	52,000

Table 1: SenseLAMA relation counts.

4 SynBERT

In this section, we cover the three steps employed to enrich LMs with sense embeddings: 1) Represent word senses from internal states; 2) Map and add sense embeddings to the LM's vocabulary; 3) Adapt cloze-style assertions and predictions to extract grounded triples. See Figure 1 for an overview. Throughout this work, we use BERT-Large (Devlin et al., 2019) as our reference LM.

4.1 Sense Representation

For representing word senses with LMs, we follow Loureiro et al. (2022) and learn sense embeddings as centroids of contextual embeddings from sense-annotated corpora and glosses. We follow the recommendation of representing contextual embeddings with weighted pooling from all layers, using weights specific to the sense matching profile (i.e., LMMS SP-USM). We also average the embeddings from annotations with gloss embeddings (centroids of contextual embeddings for lemmas and tokens in each synset's gloss). These sense embeddings are derived from a LM's frozen parameters, relying exclusively on modeling capability learned during pretraining. These sense embeddings have demonstrated state-of-the-art performance across several sense-related tasks, without bias towards most frequent senses, as observed with fine-tuning approaches (Loureiro et al., 2021).

4.2 Mapping and Injecting Embeddings

Poerner et al. (2020) found that linear mapping was sufficient for high accuracy alignment between static embeddings (unrelated to BERT) and BERT's vocabulary embeddings. We follow this approach since our sense embeddings are derived from BERT, making alignment theoretically more straightforward. In order to learn the linear mapping (using least-squares), we need tokens represented in both the LM's vocabulary embedding space (i.e., input-space) and the alternate space defined by the weighted pooling of layers used to represent the sense embeddings. We obtain this by finding tokens in the LM's vocabulary with more than 100 occurrences in Wikitext (Merity et al., 2016) and applying the same pooling used for sense representation to learn embeddings for those tokens in the alternate space. After mapping, sense embeddings are added to the LM's vocabulary as special tokens, represented using a distinct format (<WN:*synset*>) similarly to Schick and Schütze (2020).

	IsA	Desires	MadeOf
BERT	A mouse is a kind of [MASK] . animal, rabbit, cat	A mouse wants to [MASK] . play, eat, talk	A mouse is made of [MASK] . wood, clay, bone
SynBERT	mouse-eared_bat^1_n, mouser^1_n, rabbit_ears^2_n	die^2_v, forage^2_v, feed^7_v	redwood^1_n, wood^1_n, yellowwood^1_n
SynBERT	A mouse^1_n is a kind of [MASK] . rat^1_n, mouse-eared_bat^1_n, pocket_rat^1_n	A mouse^1_n wants to [MASK] . forage^2_v, feed^7_v, die^2_v	A mouse^1_n is made of [MASK] . round_bone^1_n, bone^1_n, leg^2_n
SynBERT	A mouse^4_n is a kind of [MASK] . computer_keyboard^1_n, computer_accessory^1_n, computer_memory_unit^1_n	A mouse^4_n wants to [MASK] . move$^{13}_v$, move$^{12}_v$, think^6_v	A mouse^4_n is made of [MASK] . fluorocarbon_plastic^1_n, glass^1_n, wire_glass^1_n
SynBERT	A mouse^3_n is a kind of [MASK] . dummy^1_n, shy_person^1_n, small_person^1_n	A mouse^3_n wants to [MASK] . shop_talk^1_n, talk^3_n, talk^2_n	A mouse^3_n is made of [MASK] . redwood^1_n, ironwood^2_n, yellowwood^1_n

Table 2: Top-3 masked predictions targeting 'mouse' using templates corresponding to the *IsA*, *Desires* and *MadeOf* relations. First row does not use special synset tokens in the input and shows predictions using BERT (ignoring stopwords) as well SynBERT (ignoring regular tokens). Next rows show predictions using special tokens corresponding to the 3 senses for 'mouse' available in WordNet. Their definitions are the following: mouse^1_n - any of numerous small rodents typically resembling diminutive rats [...]; mouse^4_n - a hand-operated electronic device that controls the coordinates of a cursor on your computer screen [...]; mouse^3_n - person who is quiet or timid.

4.3 Extracting Triples

Triples are extracted using assertions relating a grounded word sense with a masked tail term (e.g., [pen^1_n, *UsedFor*, ?] → "A <WN:*pen.n.01*> can be used for [MASK]."). Regular tokens are discarded from the LM's masked predictions[6], and softmax normalization is performed after filtering so that prediction scores are distributed exclusively over grounded word senses. Our default setup[7] prepends assertions with the head term's gloss from WordNet for improved results (i.e., "<WN:*synset*> can be defined as : *gloss* . [SEP] *assertion*"). We refer to Table 2 for example predictions.

5 Experiments

In this section, we explore two applications for our method that motivated this work: 1) Evaluating commonsense knowledge learned during LM pretraining; 2) Extracting precise CKGs from LMs enriched with grounded word senses.

5.1 Probing with SenseLAMA

The SenseLAMA probe described in section 3 is used to evaluate the commonsense knowledge learned while pretraining LMs, through the adaptation described in section 4. The prediction methodology described in subsection 4.3 is used to obtain ranked predictions for tail terms masked in SenseLAMA. Performance is evaluated using ranking metrics, namely mean Precision @ k and Mean Reciprocal Rank (MRR). As with LAMA, many instances admit various possible answers (1 to N). Therefore P@10 may be considered more representative of actual performance than P@1.

The complete results in Table 3 show that performance varies substantially by source and relation. It is interesting to note that for core synsets (i.e., frequent concepts), we find P@10 above 30% for most relations. Particular relation groups, such as *Holonym (Part)*, *P361 (Part of)* and *PartOf* show particularly high results (above 60% P@10), suggesting that extraction for these relation types could be reliable enough for some applications.

The Full set appears much more challenging, which is to be expected considering the 20x increase for the search space in this setting, along with several instances targeting rare concepts. While this setting is much less reliable, we still find that most relations can be accurately predicted from the top 1% of candidates (> 60% P@1000).

Out of 39 relations (Full set), the most challenging belong to ConceptNet, particularly *ReceivesAction*, *Desires*, *CausesDesire* and *HasSubevent*, supporting the claim that commonsense relations are harder to model by LM than lexical relations.

5.2 Commonsense Knowledge Extraction

While it is possible to use the method presented in this work to exhaustively query LMs and rank predictions for every synset and relation, we take a simpler approach in this experiment. Considering that the ConceptNet subset of SenseLAMA includes higher quality assertions (not generated by templates), we use these to generate new query assertions by replacing the head terms with their

[6]The head term is also removed from predictions.
[7]See subsection 6.1 for an ablation analysis.

	Core (4,960 candidates)					Full (117,659 candidates)					
	P@1	**P@3**	**P@10**	**P@100**	**MRR**	**P@1**	**P@3**	**P@10**	**P@100**	**P@1000**	**MRR**
All	24.41	40.56	59.10	83.20	35.64	7.18	13.78	23.09	45.75	71.75	12.55
WordNet	31.25	49.80	69.10	87.82	43.46	7.78	14.75	24.26	46.39	71.84	13.34
Hypernym	29.04	45.96	66.15	86.10	40.77	8.31	17.24	30.77	59.17	82.74	15.65
Holonym (Member)	42.31	69.23	88.46	100.00	57.80	1.75	3.04	5.03	13.98	41.89	3.00
Holonym (Part)	34.48	60.69	80.69	92.41	50.20	13.97	25.93	40.63	67.89	88.15	22.91
Antonym	37.94	58.16	74.11	91.49	50.09	8.10	13.72	20.96	40.56	70.32	12.55
Hypernym (Instance)	-	-	-	-	-	9.19	18.09	30.08	61.24	86.45	16.35
Meronym (Substance)	43.75	81.25	81.25	100.00	59.14	2.43	6.23	12.46	33.13	65.50	6.00
WikiData	16.18	33.09	49.26	79.41	27.62	5.05	10.12	18.83	43.91	72.07	9.69
P31 (Instance of)	10.26	23.08	23.08	61.54	16.94	2.90	6.74	13.61	37.77	68.56	6.67
P361 (Part of)	15.56	35.56	62.22	82.22	30.26	8.71	16.17	27.21	55.89	79.37	14.86
P366 (Use)	14.81	25.93	48.15	88.89	24.80	4.06	9.70	19.27	42.07	64.74	9.00
P186 (Made from)	33.33	46.67	60.00	86.67	41.66	8.61	12.83	23.63	46.64	72.77	13.03
P461 (Opposite of)	20.00	60.00	80.00	100.00	43.91	8.98	18.36	30.34	60.28	81.24	16.35
P737 (Influenced by)	-	-	-	-	-	2.53	6.33	10.76	31.33	71.20	5.85
P2283 (Uses)	-	-	-	-	-	4.85	8.58	14.93	37.31	66.42	8.42
P463 (Member of)	-	-	-	-	-	1.64	2.73	15.30	40.44	86.34	5.51
P1535 (Used by)	-	-	-	-	-	0.66	3.97	11.92	41.06	72.85	4.53
P279 (Subclass of)	-	-	-	-	-	6.06	12.12	21.21	45.45	72.73	10.64
ConceptNet	13.86	25.87	43.51	75.78	23.38	4.55	9.88	18.07	42.11	70.06	9.23
AtLocation	14.02	25.91	46.95	79.27	24.24	4.98	10.56	19.82	45.82	76.10	10.09
UsedFor	7.41	16.67	36.42	75.93	16.04	3.18	8.17	15.13	38.88	69.59	7.48
IsA	27.50	43.33	62.50	87.50	38.56	7.42	13.67	27.34	59.38	83.59	13.61
Causes	5.26	23.68	34.21	65.79	16.56	2.68	6.25	12.05	27.68	54.91	5.84
HasSubevent	3.51	14.04	19.30	43.86	10.08	0.98	2.44	5.37	14.63	35.12	2.64
HasPrerequisite	4.00	16.00	26.00	78.00	13.16	3.64	8.48	13.94	41.21	71.52	7.35
HasProperty	4.26	14.89	38.30	76.60	14.65	2.55	5.10	9.55	29.30	63.69	5.21
CapableOf	8.33	18.75	33.33	54.17	16.09	2.44	7.32	13.82	30.08	51.22	6.48
MotivatedByGoal	29.73	51.35	67.57	89.19	43.59	6.73	20.19	29.81	63.46	80.77	15.66
HasA	16.67	27.08	41.67	81.25	24.35	11.34	16.49	23.71	48.45	79.38	16.07
PartOf	36.36	54.55	75.76	90.91	48.35	10.00	21.25	37.50	66.25	87.50	19.33
CausesDesire	0.00	7.14	28.57	78.57	7.61	0.00	1.92	5.77	30.77	65.38	2.44
ReceivesAction	0.00	0.00	10.53	31.58	3.43	0.00	0.00	0.00	6.82	20.45	0.19
MadeOf	44.44	50.00	61.11	100.00	51.52	9.52	30.95	33.33	47.62	80.95	19.34
Desires	7.69	15.38	23.08	46.15	12.15	0.00	3.57	3.57	25.00	46.43	1.92
CreatedBy	-	-	-	-	-	0.00	0.00	11.76	35.29	82.35	3.43
HasFirstSubevent	-	-	-	-	-	7.14	7.14	14.29	35.71	78.57	9.23
HasLastSubevent	-	-	-	-	-	0.00	0.00	16.67	33.33	58.33	2.38

Table 3: Complete results on the SenseLAMA probing task using BERT Large with LMMS SP-USM sense embeddings. Reporting Precision at k (P@k) and Mean Reciprocal Rank (MRR). Sorted by P@1 on the Full set.

co-hyponyms. This approach also reduces the chances of generating non-sensical queries (e.g., "A <WN:*pen.n.01*> desires [MASK]."), which would result from combinatorial generation. Keeping in mind that there is likely more than a single valid prediction for each assertion, we use a threshold to extract multiple triples from each assertion's prediction distribution. This threshold is automatically determined as the median score assigned to correct predictions on the SenseLAMA (Full) probe.

This process generates 36,505 query assertions and 23,088 novel[8] triples scoring above the threshold. This novel CKG, grounded on WordNet, covers 18 commonsense relations and reaches 9.2% of all synsets. See Appendix B for detailed statistics.

6 Analysis

The analyses reported in this section focus on the following comparisons: 1) Alternatives for representing triples as cloze-style assertions; 2) Verbalization against Nearest Neighbors; 3) Mapping embeddings or retaining geometry.

6.1 Triple Representation

For this analysis, we compare alternatives for representing triples as masked assertions, specifically using synsets (special tokens) instead of regular tokens (i.e., most frequent lemma[9]) and glosses (averaged with sense embeddings and prepended to the assertion). We also combine lemmas and synsets using the *slash* representation (Schick and Schütze, 2020), where the head term in the assertion is replaced with "*lemma* / <WN:*synset*>".

Results in Table 4 show that the synset representation is more effective than lemmas, and while the combination of lemmas and synsets is better, using synsets exclusively provides slightly improved results. Glosses appear to have a substantial impact under all settings, but averaging gloss embeddings and prepending glosses shows the best results. These results also show that ConceptNet (CN) is not only the most challenging subset but also the least sensitive to these experimental choices. For completeness, Appendix C reports complete SenseLAMA results using sense embeddings (from annotated text) that have not been merged with gloss embeddings or used assertions prepended with the gloss for the head synset.

[8]Not part of the triples in SenseLAMA or its sources.

[9]Each synset may be associated to multiple lemmas. Frequencies obtained from wordfreq (Speer et al., 2018).

Token		Gloss		WN	WD	CN	ALL
Lem	Syn	Avg	Pre				
✓				19.74	17.32	16.52	18.49
✓		✓		20.79	16.62	17.40	19.40
✓			✓	36.70	27.12	22.33	31.19
✓		✓	✓	40.82	**29.79**	23.18	34.09
	✓			26.46	19.46	17.67	23.04
	✓	✓		30.39	20.72	18.60	25.78
	✓		✓	38.76	26.83	22.56	32.49
	✓	✓	✓	**43.44**	27.59	**23.38**	**35.63**
✓	✓			25.47	19.65	18.08	22.59
✓	✓	✓		27.44	20.70	19.13	24.20
✓	✓		✓	39.14	26.49	21.66	32.39
✓	✓	✓	✓	42.31	28.15	21.97	34.47

Table 4: MRR on SenseLAMA (Core) when representing lemmas (Lem) and/or synsets (Syn); averaging gloss embedding (Avg) and/or prepending the gloss (Pre).

6.2 Impact of Verbalization

Dufter et al. (2021) showed that a nearest neighbors (kNN) baseline using static embeddings could outperform BERT on the LAMA probe under comparable settings. We run a similar experiment to verify whether the same conclusion may apply to our SenseLAMA probe and SynBERT model.

In our case, the sense embeddings (mapped) added to BERT's vocabulary can be used as static embeddings. Bearing in mind that some sense embeddings from LMMS are inferred from hypernymy relations in WordNet (17.1% of senses, mostly rare), we also experiment with another set of BERT-based sense embeddings which are not derived from any relations (ARES, Scarlini et al., 2020). For a fair comparison, we do not prepend glosses for LM predictions.

As such, Table 5 reports results using kNN with sense embeddings, alongside using SynBERT with the verbalized queries (i.e., masked assertions) provided with SenseLAMA. We verify that kNN can outperform SynBERT under these conditions, but only for the more lexical-oriented relations in WordNet. For ConceptNet, the source most strictly related to commonsense knowledge, we find verbalized queries provide a clear advantage over kNN. To a lesser extent, the encyclopedic triples of WikiData are also more accurately predicted with SynBERT. This finding is in line with previous work comparing relational knowledge in BERT (Bouraoui et al., 2020).

	WordNet			WikiData			ConceptNet			All		
	P@1	**P@10**	**MRR**	**P@1**	**P@10**	**MRR**	**P@1**	**P@10**	**MRR**	**P@1**	**P@10**	**MRR**
Distance-based (kNN)												
ARES	17.87	58.91	31.07	9.56	36.76	18.27	3.97	20.45	9.31	12.58	44.31	22.80
LMMS SP-USM	**26.12**	**63.18**	**38.19**	8.09	34.56	16.67	3.10	17.44	7.97	**17.16**	**45.71**	**26.53**
Template-based (LM)												
ARES	19.86	46.96	29.08	**13.24**	**38.97**	**21.94**	9.69	36.05	17.63	15.97	42.74	24.71
LMMS SP-USM	21.17	49.57	30.39	12.50	37.50	20.72	**10.17**	**36.14**	**18.60**	16.89	44.27	25.78

Table 5: Performance comparison on SenseLAMA (Core) using the baseline kNN distance-based method (ignores relation) and the masked LM template-based method (verbalizes relation). For fair comparison, gloss prepending (see subsection 4.3) is not used for LM results.

6.3 Degradation from Mapping

Our SynBERT model features sense embeddings that result from the straightforward linear mapping of embeddings pooled from all layers into the vocabulary embedding space (see subsection 4.2).

For this analysis, we estimate the performance impact of this mapping procedure by comparing the performance of mapped and unmapped LMMS sense embeddings on the kNN baseline for SenseLAMA (described on subsection 6.2).

Results on Table 6 show that while the procedure is simple, mapped embeddings retain very similar performance to their original versions, with around 5% degradation on P@1, P@10, and MRR.

	P@1	**P@10**	**MRR**
Original	17.16	45.71	26.53
Mapped	16.21 (-5.5%)	44.21 (-3.3%)	25.31 (-4.6%)

Table 6: Performance comparison on SenseLAMA (Core) using kNN with original and mapped LMMS SP-USM embeddings.

7 Future Work

This paper focuses on BERT and WordNet due to their popularity (particularly w.r.t. probing). Future work should consider additional LMs and representations in languages other than English.

Although commonsense knowledge should remain mostly unchanged over time, the sense representations introduced in SynBERT are limited to the release date of WordNet v3.0 (2006). As noted by Eyal et al. (2022), novel concepts that have become mainstream (e.g., covid) are missing from WordNet, potentially limiting downstream applications of models such as SynBERT.

8 Conclusion

In this work, we have shown that sense embeddings, learned from grounded ontologies, can be integrated into pretrained LMs, allowing for a more precise and extensive probing of commonsense knowledge learned during pretraining compared to prior work such as LAMA.

The proposed SynBERT model, adapted from BERT, along with SenseLAMA, our new probing task grounded on WordNet, provide clearer insights into which commonsense relations are best understood by LMs, and how the commonsense domain compares against more lexical or encyclopedic knowledge. We also explore how SynBERT, or similar models, can be used to extract novel CKGs which may support recent hybrid methods fusing CKGs and LMs (e.g., Zhang et al., 2022), or enable symbolic-first methods (e.g., Huang et al., 2021) to leverage precise commonsense knowledge learned without supervision by LMs.

Reproducibility

SenseLAMA (Core and Full sets), SynBERT and related code is freely available at `https://github.com/anonymous/synbert` (MIT License). SynBERT, with the full set of 117k synsets, contains 454M parameters.

References

Carlos Aspillaga, Marcelo Mendoza, and Alvaro Soto. 2021. Tracking the progress of language models by extracting their underlying knowledge graphs.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, pages 4661–4672, Online. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *AAAI*.

Jordan Boyd-Graber, Christiane Fellbaum, Daniel Osherson, and Robert Schapire. 2005. Adding dense, weighted connections to wordnet. In *GWC 2006*, GWC 2006: 3rd International Global WordNet Conference, Proceedings, pages 29–35. Masaryk University. 3rd International Global WordNet Conference, GWC 2006 ; Conference date: 22-01-2006 Through 26-01-2006.

Ruben Branco, António Branco, João António Rodrigues, and João Ricardo Silva. 2021. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yohan Chalier, Simon Razniewski, and Gerhard Weikum. 2020. Joint reasoning for multi-faceted commonsense knowledge. In *Automated Knowledge Base Construction*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Bhavana Dalvi Mishra, Niket Tandon, and Peter Clark. 2017. Domain-targeted, high precision knowledge extraction. *Transactions of the Association for Computational Linguistics*, 5:233–246.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter, Nora Kassner, and Hinrich Schütze. 2021. Static embeddings as efficient knowledge bases? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2353–2363, Online. Association for Computational Linguistics.

Matan Eyal, Shoval Sadde, Hillel Taub-Tabib, and Yoav Goldberg. 2022. Large scale substitution-based word sense induction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4738–4752, Dublin, Ireland. Association for Computational Linguistics.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

Catherine Havasi, Robyn Speer, James Pustejovsky, and Henry Lieberman. 2009. Digital intuition: Applying common sense using dimensionality reduction. *IEEE Intelligent systems*, 24(4):24–35.

Jiani Huang, Ziyang Li, Binghong Chen, Karan Samel, Mayur Naik, Le Song, and Xujie Si. 2021. Scallop: From probabilistic deductive databases to scalable differentiable reasoning. In *Advances in Neural Information Processing Systems*, volume 34, pages 25134–25145. Curran Associates, Inc.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Stanislaw Jastrzębski, Dzmitry Bahdanau, Seyedarian Hosseini, Michael Noukhovitch, Yoshua Bengio, and Jackie Cheung. 2018. Commonsense mining as knowledge base completion? a study on the impact of novelty. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 8–16, New Orleans, Louisiana. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Donald Ervin Knuth, James H. Morris, and Vaughan R. Pratt. 1977. Fast pattern matching in strings. *SIAM J. Comput.*, 6:323–350.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany. Association for Computational Linguistics.

Daniel Loureiro, Alípio Mário Jorge, and Jose Camacho-Collados. 2022. Lmms reloaded: Transformer-based sense embeddings for disambiguation and beyond. *Artificial Intelligence*, 305:103661.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, 47(2):387–443.

John P. McCrae and David Cillessen. 2021. Towards a linking between WordNet and Wikidata. In *Proceedings of the 11th Global Wordnet Conference*, pages 252–257, University of South Africa (UNISA). Global Wordnet Association.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

George A. Miller. 1992. Wordnet: A lexical database for english. *Commun. ACM*, 38:39–41.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. 2021. Advanced semantics for commonsense knowledge extraction. *Proceedings of the Web Conference 2021*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA,*

January 27 - February 1, 2019, pages 3027–3035. AAAI Press.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Online. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. Luminosoinsight/wordfreq: v2.2.

Shane Storks, Qiaozi Gao, and Joyce Y. Chai. 2019. Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches. *CoRR*, abs/1904.01172.

Vinitra Swamy, Angelika Romanou, and Martin Jaggi. 2021. Interpreting language models through knowledge graph extraction. In *eXplainable AI approaches for debugging and diagnosis.*

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Niket Tandon, Gerard de Melo, Fabian Suchanek, and Gerhard Weikum. 2014. Webchild: Harvesting and organizing commonsense knowledge from the web. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, page 523–532, New York, NY, USA. Association for Computing Machinery.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Peifeng Wang, Filip Ilievski, Muhao Chen, and Xiang Ren. 2021. Do language models perform generalizable commonsense inference? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3681–3688, Online. Association for Computational Linguistics.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140, Online. Association for Computational Linguistics.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. Symbolic knowledge distillation: from general language models to commonsense models. *CoRR*, abs/2110.07178.

Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. 2021. Fusing context into knowledge graph for commonsense question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1201–1207, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. GreaseLM: Graph REASoning enhanced language models. In *International Conference on Learning Representations*.

A Templates

Templates used for WordNet and WikiData triples are available in Table 7. In order to make predictions more consistent across sources, we found the most frequent determiners used with the head and tail terms of ConceptNet assertions and applied them on the WordNet and WikiData queries, wherever those same head and tail terms occurred.

B Extraction Statistics

Table 8 reports the relation counts for triples extracted using the procedure described on subsection 5.2.

C SenseLAMA without gloss information

Table 9 reports results by relation on SenseLAMA (Full) when not prepending glosses or averaging sense embeddings with gloss embedding. This is intended to better demonstrate which relations are most affected by the use of glosses, besides their overall impact on this probing task.

Source	Relation	Template	Example Pair ([H]ead, [T]ail)	
WordNet	Hypernym	[H] is a type of [T]	medicine$^{2}_{n}$	drug$^{1}_{n}$
	Holonym (Member)	[H] is a member of [T]	princess$^{1}_{n}$	royalty$^{2}_{n}$
	Holonym (Part)	[H] is part of [T]	jaw$^{1}_{n}$	skull$^{1}_{n}$
	Antonym	[H] is the opposite of [T]	straight$^{8}_{n}$	curved$^{1}_{a}$
	Hypernym (Instance)	[H] is an example of [T]	sahara$^{1}_{n}$	desert$^{1}_{n}$
	Meronym (Substance)	[H] is made of [T]	bread$^{1}_{n}$	flour$^{1}_{n}$
WikiData	P31 (Instance of)	[H] is an example of [T]	capitalism$^{1}_{n}$	political_orientation$^{1}_{n}$
	P361 (Part of)	[H] is part of [T]	regulation$^{1}_{n}$	politics$^{1}_{n}$
	P366 (Use)	[H] is used for [T]	vegetable_oil$^{1}_{n}$	makeup$^{1}_{n}$
	P186 (Made from)	[H] is made from [T]	eiffel_tower$^{1}_{n}$	wrought_iron$^{1}_{n}$
	P461 (Opposite of)	[H] is the opposite of [T]	technophilia$^{1}_{n}$	technophobia$^{1}_{n}$
	P737 (Influenced by)	[H] is influenced by [T]	mozart$^{1}_{n}$	bach$^{1}_{n}$
	P2283 (Uses)	[H] uses [T]	oil_painting$^{1}_{n}$	oil_paint$^{1}_{n}$
	P463 (Member of)	[H] is a member of [T]	taiwan$^{1}_{n}$	world_trade_organization$^{1}_{n}$
	P1535 (Used by)	[H] is used by [T]	rocket_fuel$^{1}_{n}$	rocket$^{2}_{n}$
	P279 (Subclass of)	[H] is a type of [T]	baroque$^{1}_{n}$	expressive_style$^{1}_{n}$

Table 7: Templates used to verbalize triples from WordNet and WikiData. Not required for our ConceptNet subset.

Relation	Count
IsA	6,557
AtLocation	5,104
PartOf	2,559
UsedFor	2,523
MadeOf	1,293
Causes	680
CausesDesire	663
HasPrerequisite	659
HasA	644
CapableOf	534
MotivatedByGoal	532
HasProperty	424
CreatedBy	390
Desires	229
HasSubevent	161
ReceivesAction	77
HasLastSubevent	53
HasFirstSubevent	6

Table 8: Relation counts for novel triples extracted.

	Core (4,960 candidates)					Full (117,659 candidates)					
	P@1	**P@3**	**P@10**	**P@100**	**MRR**	**P@1**	**P@3**	**P@10**	**P@100**	**P@1000**	**MRR**
All	14.39	25.85	40.62	67.04	23.04	2.71	5.36	9.73	21.68	40.25	5.07
WordNet	17.53	29.88	36.48	44.39	26.46	3.00	5.69	9.89	20.86	37.83	5.32
Hypernym	14.52	25.62	31.37	38.90	22.67	2.76	6.28	12.70	28.97	47.23	5.99
Holonym (Member)	19.23	38.46	57.69	84.62	32.02	0.39	0.68	1.61	5.60	24.05	0.87
Holonym (Part)	11.03	27.59	49.66	74.48	22.94	7.76	12.86	19.29	36.62	57.88	11.78
Antonym	34.04	48.94	64.54	78.01	44.43	3.30	6.16	9.80	19.01	30.42	5.55
Hypernym (Instance)	-	-	-	-	-	1.21	3.32	7.23	14.21	26.87	3.04
Meronym (Substance)	25.00	43.75	62.50	87.50	37.89	0.91	1.67	3.80	16.87	38.15	2.11
WikiData	10.29	22.06	36.76	63.24	19.46	1.50	3.52	8.10	22.63	46.44	3.67
P31 (Instance of)	5.13	10.26	23.08	46.15	10.51	0.67	2.26	5.19	16.81	37.30	2.31
P361 (Part of)	11.11	20.00	44.44	73.33	20.85	1.98	4.02	9.44	27.21	54.06	4.43
P366 (Use)	3.70	25.93	25.93	51.85	15.31	1.97	3.41	9.04	24.51	46.92	4.05
P186 (Made from)	26.67	33.33	46.67	80.00	34.34	1.72	3.76	9.08	25.35	53.05	4.06
P461 (Opposite of)	20.00	50.00	70.00	90.00	37.02	4.99	10.38	20.36	38.12	59.28	9.69
P737 (Influenced by)	-	-	-	-	-	0.95	2.53	4.43	14.24	43.99	2.29
P2283 (Uses)	-	-	-	-	-	1.12	2.99	8.21	20.15	44.78	3.31
P463 (Member of)	-	-	-	-	-	1.64	6.01	14.75	46.99	84.15	5.91
P1535 (Used by)	-	-	-	-	-	0.00	1.32	5.30	15.23	49.01	1.46
P279 (Subclass of)	-	-	-	-	-	1.52	1.52	3.03	22.73	40.91	2.83
ConceptNet	9.59	19.48	34.69	67.64	17.67	1.84	5.20	11.21	29.23	55.78	4.93
AtLocation	9.15	18.29	33.54	72.87	17.08	2.29	6.08	12.45	31.47	62.65	5.54
UsedFor	4.32	12.96	29.01	66.05	11.99	1.51	3.63	9.53	27.08	56.43	4.09
IsA	19.17	31.67	52.50	71.67	28.78	0.98	6.64	14.06	37.50	61.13	5.44
Causes	7.89	13.16	21.05	39.47	12.51	0.89	1.79	4.91	16.96	35.27	2.22
HasSubevent	0.00	8.77	17.54	29.82	5.88	0.98	0.98	2.44	10.24	21.46	1.49
HasPrerequisite	8.00	16.00	32.00	62.00	14.71	1.21	4.24	10.30	29.09	53.94	4.37
HasProperty	0.00	10.64	27.66	80.85	9.45	1.27	5.10	7.01	32.48	60.51	4.36
CapableOf	6.25	12.50	22.92	66.67	12.29	0.81	3.25	11.38	21.95	47.97	3.82
MotivatedByGoal	10.81	37.84	54.05	86.49	27.02	5.77	12.50	27.88	49.04	70.19	12.30
HasA	14.58	16.67	39.58	62.50	20.76	3.09	10.31	14.43	31.96	62.89	7.28
PartOf	39.39	51.52	63.64	93.94	47.47	11.25	16.25	30.00	42.50	75.00	16.43
CausesDesire	0.00	0.00	14.29	50.00	3.41	0.00	0.00	1.92	13.46	30.77	0.63
ReceivesAction	21.05	31.58	36.84	52.63	28.18	0.00	0.00	4.55	20.45	38.64	1.29
MadeOf	5.56	38.89	55.56	88.89	24.40	0.00	0.00	9.52	33.33	57.14	2.63
Desires	0.00	7.69	7.69	53.85	5.54	0.00	7.14	10.71	10.71	50.00	3.65
CreatedBy	-	-	-	-	-	0.00	11.76	11.76	29.41	70.59	5.62
HasFirstSubevent	-	-	-	-	-	0.00	0.00	0.00	50.00	85.71	2.27
HasLastSubevent	-	-	-	-	-	0.00	0.00	0.00	16.67	41.67	0.85

Table 9: Complete results on the SenseLAMA probing task using BERT Large with LMMS SP-USM sense embeddings, not averaged with gloss embeddings and without prepending glosses to assertions, in contrast to Table 3. Reporting Precision at k (P@k) and Mean Reciprocal Rank (MRR). Sorted by P@1 on the Full set.