FACULTY OF ENGINEERING OF THE UNIVERSITY OF PORTO

Lie-o-matic: using natural language processing to detect contradictory statements

Beatriz Souto de Sá Baldaia

Master in Informatics and Computing Engineering

Supervisor: Henrique Daniel de Avelar Lopes Cardoso, PhD

Co-supervisor: Carlos Manuel Milheiro de Oliveira Pinto Soares, PhD

July 31, 2020

Lie-o-matic: using natural language processing to detect contradictory statements

Beatriz Souto de Sá Baldaia

Master in Informatics and Computing Engineering

July 31, 2020

Abstract

With the faster and wider flow and production of information on media, people, such as journalists, struggle to cope with the increasing data disclosure. It is hard to monitor and verify that information that might also be corrupted (containing lies, inconsistencies, contradictions, etc.). Focusing on contradictions spread on media, they might be difficult to capture and gather, for instance, due to the time gap between them. Considering the problem above, and the constant evolution in Natural Language Processing (NLP) techniques and in machine learning, we are interested in taking advantage of those recent developments to tackle the specific NLP task of detecting contradictions in text.

This dissertation focus on transfer learning, thus we investigate the effect of resorting to different document relationships, but still related with our task of detecting contradictions, on the performance of a supervised learning classification model. Hence, we address the problem as a binary sentence-pair classification task, built on top of a pre-trained BERT model, to later predict if two texts are contradictory or not. Literature on contradiction detection has focused almost on separating antonyms and contrasting words. To the best of our knowledge, no systematic investigation has considered transfer learning for the task of contradiction detection.

To illustrate our approach, contradictions in a political domain were used as a case study. For the new document relationships under study, we collected data from five publicly available corpora: MultiNLI, US2016, Argumentative Microtext, Argument Annotated Essays, and W2E. And for the target relation to be tested, we have built two datasets containing pairs of contradictory statements from two different sources: an online article exposing Donald Trump contradictory claims, and government-related instances of the MultiNLI corpus. To evaluate the conducted experiments, we measure classification performance mainly through ROC and Precision-Recall curve analysis.

Our findings point towards the direction that other datasets, designed for different tasks, but still related with the target task, can be used to boost the inference model learning performance on our target task of detecting contradictions. We conclude that text genre, the relation of disagreement between two propositions, and the specificity/profile of a person's language have an higher positive impact. Thus, it can be helpful to select new data that incorporates these types of relationship between documents. Nonetheless, we have faced some limitations in our research, such as the lack of robustness in the target dataset built by a single and untrained annotator. Furthermore, by using BERT model, that achieves state-of-the-art results in several NLP tasks, we already obtain high classification results just by using the target dataset for both train and test sets, leaving only a small margin for improvements.

Resumo

Com o rápido e maior alcance do fluxo de informação e produção da mesma nos media, pessoas, como os jornalistas, têm dificuldade em lidar com a crescente divulgação de dados. É difícil monitorar e verificar essa informação que pode também estar corrompida (conter mentiras, inconsistências, contradições, etc.). Focando-nos em contradições divulgadas nos media, elas poderão ser difíceis de identificar e reunir, por exemplo, devido ao intervalo de tempo entre as suas ocorrências. Considerando o problema acima, e a constante evolução em técnicas de processamento de linguagem natural e *machine learning*, estamos interessados em tirar vantagens desses recentes desenvolvimentos para atacar o caso específico de deteção de contradições em texto.

Esta dissertação centra-se em *transfer learning* (transferência de conhecimento), assim investigamos o efeito de recorrer a diferentes relações entre documentos, mas ainda relacionadas com a nossa tarefa de detetar contradições, no desempenho de um modelo de classificação de aprendizagem supervisionada. Desta forma, abordamos o problema como uma tarefa de classificação binária de pares de frases, desenvolvida sob um modelo BERT pré-treinado, para depois prevermos se dois textos são contraditórios ou não. Estudos em deteção de contradições têm-se focado mais em distinguir antónimos e palavras contrastantes. Tanto quanto é do nosso conhecimento, nenhuma investigação sistemática alguma vez considerou *transfer learning* para a tarefa de detetar contradições.

Para ilustrarmos a nossa abordagem, contradições no domínio político foram usadas como caso de estudo. Para as novas relações entre documentos a serem estudadas, reunimos dados provenientes de cinco corpos disponíveis ao público: MultiNLI, US2016, Argumentative Microtext, Argument Annotated Essays e W2E. E para a relação alvo a ser testada, construímos dois conjuntos de dados contendo pares de textos contraditórios provenientes de duas origens diferentes, um artigo online expondo aclamações contraditórias de Donald Trump, e instâncias do género governamental do corpo MultiNLI. Para avaliar as experiências guiadas, medimos o desempenho da classificação maioritariamente a partir de análises à curva característica de operação (curva ROC) e à curva de Precisão-Abrangência.

Os nossos resultados apontam para que outros conjuntos de dados, desenvolvidos para diferentes tarefas, mas ainda relacionados com a tarefa alvo, podem ser usados para melhorar o desempenho de aprendizagem de um modelo de inferência sobre a nossa tarefa alvo de detetar contradições. Nós concluímos que a categoria dos textos, a relação de desacordo entre duas proposições, e a especificidade/perfil de linguagem de uma pessoa, têm maior impacto positivo. Assim, poderá ser útil selecionar novos dados que incorporem estes tipos de relações entre documentos. Não obstante, no nosso estudo enfrentamos limitações, como a falta de robustez no conjunto de dados alvo construído por um único e não treinado anotador. De referir ainda, ao usarmos o modelo BERT, que atinge resultados estado da arte em diversos problemas de processamento de linguagem natural, já obtemos resultados de classificação altos apenas usando o conjunto de dados alvo para ambos conjuntos de treino e teste, o que nos deixa com uma pequena margem para melhorias.

Acknowledgements

I would like to thank my supervisors, Professor Henrique Cardoso and Professor Carlos Soares, for all the guidance, feedback and valuable advice. And a special thanks to my family and friends for all the emotional support.

Beatriz Baldaia

“You know, it really doesn’t matter what the media write,
as long as you’ve got a young and beautiful piece of ass.”

Donald John Trump

Contents

List of Figures

List of Tables

Abbreviations

BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag Of Words
DSM	Distributional Semantic Model
ELMo	Embeddings from Language Models
FN	False Negative
FP	False Positive
ML	Machine Learning
LIWC	Linguistic Inquiry and Word Count
LM	Language Model
LSTM	Long Short-Term Memory
MLM	Masked Language Model
MLN	Markov Logic Network
NEL	Named-Entity Linking
NER	Named-Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
NSP	Next Sentence Prediction
POS	Part-Of-Speech
QA	Question Answering
RNN	Recurrent Neural Network
RTE	Recognizing Textual Entailment
SVM	Support Vector Machine
TE	Textual Entailment
TN	True Negative
TP	True Positive

Chapter 1

Introduction

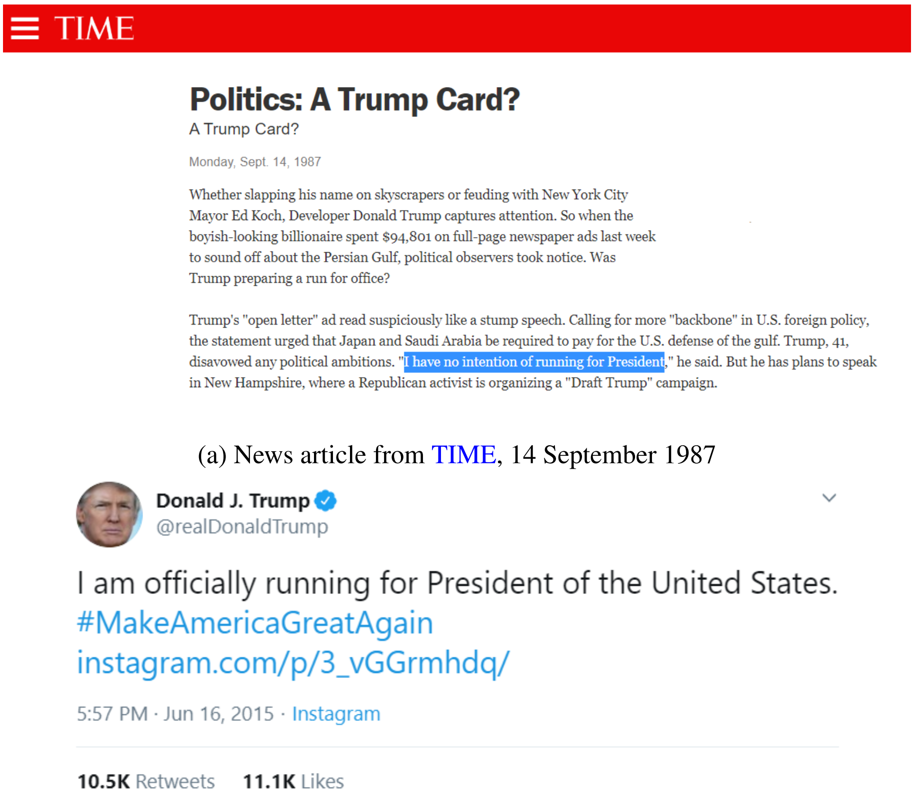
Media is a tool that gathers and delivers information. It has played an important role in our lives since the development of writing and paper, which not only enabled the preservation of data but also longer-distance communications. Currently, we are in the Information age, where the digitization of voice, image, sound and text, through the use of social networks (Al-Rawi, 2019), personal computers, mobile devices, and wearable technologies, provided the means for a faster flow of information in a wider scale. Not only the distribution and access to information is easier now, but also its production, which leads us to an environment predisposed to spreading misinformation (Allcott et al., 2019). In fact, besides the problem of having false and contradictory data, there are also cases where an entity shows a different opinion in different time-frames or situations. All these inconsistencies are hard to monitor, detect and verify due to the amount of information that we are exposed to (Cohen et al., 2011). There are professionals, namely journalists, that dedicate their time dealing with this issue (Nieminen and Rapeli, 2019), but the human being struggles to cope with the increasing data disclosure and complexity of the task. Therefore, there is a need to automate this process (Graves, 2018).

Modern techniques in Natural Language Processing (NLP), for interpreting human language, have proven to be a significant help in automating several tasks, such as fact-checking, entailment recognition and contradiction detection, that, before, were just driven by professional workers (Thorne and Vlachos, 2018). However, it faces some challenges, such as the amount of data needed, the data structure (since the language has to be in a machine-interpretable format), word meanings, and the relation between words (Sarr and Sall, 2017). Regarding capturing words meaning, a very popular approach in NLP is word embeddings that are vector representations of words, generated by Distributional Semantic Models (DSMs), capable of extracting lexical semantics (Bakarov, 2018).

A *contradiction* is a semantic relation where two sentences cannot be true simultaneously. Besides factual contradictions, where a sentence goes against common knowledge, people's interest might change over time, and so it is expectable to find contradictions in present and past speeches. These changes of beliefs can also be strategical, in order to please, manipulate or deceive. An example is when candidates, in political campaigns, change their positions regarding controversial issues (Putnam et al., 2014). Therefore, it is difficult to collect examples of real cases of contradictions, not only because of a lack of resources, but also due to their subjective nature. It is a complex task, but we believe that it can be addressed resorting to word embeddings and machine learning. For this reason, the aim of this dissertation is to tackle the problem of detecting contradictions, taking into account data limitations (the challenge of collecting contradictory statements), by using NLP and machine learning modern techniques, namely word embeddings and transfer learning, to achieve better performance in automatic detection of contradictions.

1.1 Problem Statement and Motivation

What people say in social media can be forgotten, ignored, or not given too much relevance at that time, but cannot be deleted. Data is persistent, and so there is the possibility of taking advantage of that and identify whenever someone goes against a previous claim. In Figure 1.1, we can see an example of the referred problem, where, first, Donald Trump, the president of the United States of America, asserts that he has no intention of running for President, and years later announces his application.



TIME

Politics: A Trump Card?

A Trump Card?

Monday, Sept. 14, 1987

Whether slapping his name on skyscrapers or feuding with New York City Mayor Ed Koch, Developer Donald Trump captures attention. So when the boyish-looking billionaire spent $94,801 on full-page newspaper ads last week to sound off about the Persian Gulf, political observers took notice. Was Trump preparing a run for office?

Trump's "open letter" ad read suspiciously like a stump speech. Calling for more "backbone" in U.S. foreign policy, the statement urged that Japan and Saudi Arabia be required to pay for the U.S. defense of the gulf. Trump, 41, disavowed any political ambitions. "I have no intention of running for President," he said. But he has plans to speak in New Hampshire, where a Republican activist is organizing a "Draft Trump" campaign.

(a) News article from TIME, 14 September 1987

Donald J. Trump
@realDonaldTrump

I am officially running for President of the United States. #MakeAmericaGreatAgain instagram.com/p/3_vGGrmhdq/

5:57 PM · Jun 16, 2015 · Instagram

10.5K Retweets 11.1K Likes

(b) Donald Trump tweet, 16 June 2015

Figure 1.1: Donald Trump contradicting himself on his intentions of running for president.

From the previous example, we notice that the considered information can be from different sources (social network services, online magazines, etc.), distinct timelines (time gap between two contradictions), and of various formats (answers to interviews, quotes in news articles' body, posts). It is indisputable that the recognition of contradictions in this case is a challenging task, as it requires an analytical ability and, mostly, access to previous data (i.e. given the statement in Figure 1.1b, the reader would need to be already aware of the past claim, Figure 1.1a, to acknowledge the first as contradictory). Thus, it is also hard for a person to collect pairs of real-world examples of contradictory texts.

We rely on computers for helping in several natural language processing tasks, such as email filtering (spam, primary, social, and promotion filters), smart assistants (e.g., Apple Siri and Amazon Alexa), and language translation. Recent machine techniques for natural language processing have been very successful in many tasks, and are also promising for contradiction detection, already with some good results. Nevertheless, it is hard to collect data, so it would be useful if knowledge from other related problems could be used to improve a model performance on contradiction detection. Such reuse of knowledge can be achieved through *Transfer Learning*. Therefore, the main aim of our study is to propose a model for automatic detection of contradictions, while exploring and exploiting other related tasks for improving the model learning performance. Hence, this approach can be of practical relevance for detecting contradictions in domains that lack resources and data for that purpose, and speculate patterns of contradictory discourses and similar relations.

1.2 Research Questions and Objectives

Taking into account the aim of our study, we formulate the following two research questions:

Considering our target task of detecting whether two documents are contradictory or not, ...

$\mathscr{Q}_1$. *... can a classification model be effective when only trained with examples whose document-pair relations are different from the target one (contradictions)?*

$\mathscr{Q}_2$. *... can other examples, that incorporate document-pair relations different from the target one, be used to provide an extra training set of contradictory statements, in order to improve a model learning performance?*

The objectives, acting as milestones toward the main aim of the study, are as follows:

- Based on the revised researches, formulate a set of assumptions and hypotheses that will serve as basis for the development of the proposed system;
- Select document-pair relationships that might be related with our target relation, contradiction between two texts;
- Select existing corpus that can be applied to our research purpose;
- Collect and prepare own data, and construct our datasets;

- Develop and implement a method for detecting contradictions between two texts;
- Evaluate the system on real cases (gathered data);
- Measure and compare models performances.

1.3 Document Structure

This dissertation structure consists of six chapters, followed by Appendix and References. Next, we list a brief description of the remaining document structure:

1. Related Work: Covers the main concepts for this study, focusing on machine learning for supervised classification, transfer learning, numerical representation of text, and detection of contradictions. Also addresses previous works on language pattern analysis.
2. Methodology: Introduces language patterns, in general and in a political context, and describes the approach to accomplish the main aim of this study.
3. Data and Experimental Setup: Presents the datasets created for this study, the experimental environment and experiments' details.
4. Results and Discussion: Shows the results obtained from the conducted experiments, and contains the analysis of those results, through evaluation measures, and conclusions drawn from those analyses.
5. Conclusions and Future Work: Summarizes the findings, answers the research questions, considers our research's scientific contributions, outlines the limitations of the final product, and gives insights for further research and areas to improve.

Chapter 2

Related Work

In this chapter we present the background and definitions of essential concepts, in the context of this project. We start by introducing, in Section 2.1, Natural Language Processing and some of the different existing applications in this field. Section 2.2 presents supervised learning, focusing on classification tasks. Thus, we provide important definitions, such as input representation, the two types of output, hard and soft classification, inference methods' algorithms, and metrics used for evaluation. Section 2.3 talks about transfer learning, its advantages and categories, and the types of knowledge that can be transferred. Section 2.4 exposes the evolution of numerical vector representation of text, distinguishing the two main types of text representation: localist and distributed. Furthermore, we complement this section by giving examples of techniques used for each type of text representation. We open section 2.5 with the definition of *"contradiction"*, and then proceed to present existing approaches addressing the problem of contradiction detection. Moreover, we give an overview of datasets containing examples of contradictions that, therefore, can be used for tackling this problem. Finally, section 2.6 covers the topic of language patterns, demonstrating different approaches held in order to analyse which are the prominent linguistic markers in deceptive texts and how they can be used to help to predict the truthfulness of a text.

2.1 Introduction to Natural Language Processing

Natural language processing (NLP) uses a set of computational techniques with the purpose of processing, understanding, and producing human language content (Hirschberg and Manning, 2015). Below we list some of the several NLP tasks and applications:

- Automatic summarization: Shortening a set of data, creating a subset which represents the most relevant contents (Tas and Kiyani, 2017).

- Named-entity linking (NEL): Assign a unique identity to entities (e.g, a person or a location) present in a text (DAI et al., 2012).

- Named-entity recognition (NER): Identify and classify entities mentioned in a text into specific classes (e.g, person names, locations, dates, quantities) (Nadeau and Sekine, 2007).

- Machine translation: Using corpus statistical, and neural techniques to translate text or speech from one language to another, handling existing variations in linguistic typology and idioms (Chu and Wang, 2018).

- Part-of-speech tagging (POS tagging): Classify words, as nouns, verbs, adjectives, adverbs, etc, considering its definition and context (relationship with adjacent and related words in a text) (Kanakaraddi and Nandyal, 2018).

- Question answering (QA): Being able to automatically answer questions made by humans (Mishra and Jain, 2016).

- Textual entailment (TE): Check whether the truth of one text fragment is followed from another text. Therefore, a termed text entails an hypothesis if who is reading the text can infer that the hypothesis is most likely true (Androutsopoulos and Malakasiotis, 2010).

NLP faces several challenges which can be related with the structure of the data used, with the amount of existing data considered for a task, and with data semantics. Regarding data, we might deal with: limited available data, for instance, fewer resources for less popular languages (Das et al., 2016); languages with complex linguistic structure, such as Arabic (Farghaly and Shaalan, 2009) and Chinese (Ouyang et al., 2019); and dealing with large texts or multiple document sources, which is harder to extract and process their contexts (e.g., multi-document summarization (Hahn and Mani, 2000)). In natural language understanding, ambiguity is, perhaps, the main problem in NLP as words meaning changes over contexts (Hussein, 2018). There is also the issue of synonymy, where the same idea can be expressed through the use and conjugation of different terms, depending on the context (Tovar et al., 2018); co-reference which aims at identifying all expressions referring to the same entity (Sukthanker et al., 2020); and author's emotion and subjectivity in statements, tackled by sentiment analysis (Khan et al., 2016).

2.2 Supervised Learning and Classification

In this section we will focus on supervised learning applied to classification tasks that are, indeed, the type of tasks in which this learning method is usually employed.

Supervised learning, perhaps the most used type of machine learning in natural language processing applications, aims at learning a function, over a set of labeled training examples, that for an input (data instance) will predict its output (class/label).

Regarding the input, a data instance (d) is represented as a features vector ($\vec{d} = \langle t_1, ..., t_n \rangle$), and its values ($t_1, ..., t_n$) depend on the chosen data representation scheme. As for the output, we may have a multi-label classification (more than one possible label for an instance) or a single-label classification, where we test if an instance belongs to the class of interest, the positive class; otherwise, is predicted as negative.

The classification function can perform a hard classification, where the result is a value in a discrete set, or a soft classification which predicts a value that can range over real values in interval

[0, 1] (Liu et al., 2011). The soft approach can be seen as ranking a class, saying how likely it is for the instance to belong to that class. Furthermore, a soft classification can be converted into a hard one by thresholding.

Supervised learning involves splitting the data set into training set and test set (and also validation set if parameter tuning is going to be conducted). The data used in training should not be used for testing, in order to prove that the induced classification function generalises to unseen data (avoiding overfitting in the train data).

The algorithms used for the inference methods can be numeric (the output is a numeric score) or symbolic (usually following a hard classification, the output is directly a class label) (Honavar, 1995). Two well known examples of approaches that can be used as symbolic inference methods, also referred to as discriminant functions, are:

- Decision Trees: The features are discrete values, the test sequence is encoded as a tree structure (logical tree where the nodes represent conditions and conclusions, and the branches decisions), and each classification conducts a binary test. An application example is a diagnostic Chatbot for supporting primary health care systems (Kidwai and RK, 2020), that uses decision tree algorithm to help building, through a top-down approach, a diagnosis of a user's condition based on its symptoms.

- Support Vector Machines (SVMs): Considering a n-dimensional space, where n is the number of input features, a hyperplane divides that space into two sides, one for the positive examples and the other for the negative ones. If the instances are not linearly separable, a technique called "kernel trick" is used to transform the low-dimensional input space into a higher-dimensional space, so the classes can then be separated. An application example is a novel two-pass classifier architecture (Padmavathy et al., 2020), combining SVM and artificial neural network, that infers the drug-satisfaction level of patients who have tried it.

A common numeric alternative is probabilistic classifiers, that output an estimation of the conditional probability of an instance belonging to a certain class. A Naïve Bayes classifier is one of those. There are different Naïve Bayes classifier variants, all based on applying Bayes' theorem that assumes that the input features, $\vec{d} = \langle t_1, ..., t_n \rangle$, are independent of each other, given the class variable, c, as shown in Equation 2.1:

$$P(\vec{d} \mid c) = \prod_{k=1}^{|T|} P(t_k \mid c) \tag{2.1}$$

An application example of this probabilistic classifier is a machine learning approach to classify a person's depression level based on its social media posts (on Facebook and Twitter), resorting to Naïve Bayes algorithm to determine whether the sentiment in a post is positive, negative or neutral (Asad et al., 2019).

In the probabilistic approach, the output relative magnitude can be seen as the degree of confidence the classifier has in its prediction results.

Table 2.1: Confusion matrix for binary classification. Positive is the class of interest.

		Predicted values	
		Positive (P)	Negative (N)
Actual values	Positive (P)	True Positive (TP)	False Negative (FN)
	Negative (N)	False Positive (FP)	True Negative (TN)

In natural language processing tasks, evaluation is usually done experimentally, rather than analytically, due to the subjectivity of the task. In experimental evaluation we measure the classifier effectiveness based on its ability to make correct predictions. Through a confusion matrix, we can summarize the performance of a classification model. This $N * N$ matrix, being N the number of classes, presents the cases of each distinguish class against the model prediction results for those examples. Now we will focus on metrics computed considering a binary classification, thus, Table 2.1 represents a confusion matrix in that situation.

The values in Table 2.1 (TP, FP, FN, and TN) are used in order to calculate measures for evaluating a classifier, such as the ones listed bellow:

Precision

Measures the proportion of right predictions when considering only the cases which the model classified as belonging to class of interest (positive class). A high precision value means that an example classified as positive is, indeed, more likely to be positive. It can be defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

Recall (or sensitivity)

Measures the proportion of positive examples that are recognized as so. A high recall value means the class of interest is often correctly recognized. It can be defined as follows:

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

Accuracy

Measures the frequency of the model's correct predictions, among all the ran predictions. Accuracy faces the problem of assuming equal costs for both types of errors, FP and FN, and can be misled when resorting to an unbalanced data set. For example, if we have an unbalanced data set, consisting only of 1% positive examples, we can have all the positive instances wrongly labelled as negatives and still achieve an accuracy of 99% (when all instances are classified as negative). It can be defined as follows:

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{2.4}$$

Fallout

Measures the proportion of non-targeted items (negative instances) that were mistakenly selected (predicted as belonging to the class of interest). It can be defined as follows:

$$Fallout = \frac{FP}{FP+TN} \tag{2.5}$$

Receiver Operating Characteristic (ROC curve)

Measures how different levels of fallout influence recall. Therefore, it is created by plotting the true positive rate (recall, Equation 2.3) against the false positive rate (fallout, Equation 2.5), varying the threshold value.

Precision and recall measure how well a classifier identifies the class of interest, and both should not be considered isolated (Alvarez, 2002). Thus, there are two measures that combine precision and recall: breakeven point and F functions. The breakeven point is the value at which precision equals recall. It can be determined by plotting the precision-recall curve and a bisecting line, and register the point where both intersect. F-measure (or F-score), assigns a degree of importance to precision and recall. Indeed, that degree is represented by the factor β $(0 \leq \beta \leq \infty)$, meaning that recall is β times as important as precision. The general formula for F-measure is:

$$F_\beta = (1+\beta^2) * \frac{Precision * Recall}{(\beta^2 * Precision) + Recall} \tag{2.6}$$

When β is equal to 1, precision and recall assume the same importance, and F-measure (in this case, F_1 score) is seen as the their harmonic mean.

2.3 Transfer Learning

In machine learning, the training and testing data are usually in the same feature space and have the same distribution. However, when there is no sufficient good and balanced data for training and testing, both data sets might be in a different feature space, or with different distributions, in order to improve the model learning process, and avoid the extra, costly and exhausting work of data labeling, or even the need of re-collecting data. In this case we are resorting to transfer learning (Pan and Yang, 2010), where we consider training and testing sets from distinguish domains, tasks, and distributions, and take advantage from previously learned knowledge (from a source task) to solve new problems (the target task) and boost performance. Therefore, in contrast to traditional machine learning that learns a task from scratch, transfer learning passes the knowledge from a source task to a target task when the latter struggles in lacking training data quality.

Considering the variations in source and target tasks and domains, transfer learning can be divided into three categories:

- Inductive transfer learning: The target and source tasks are different and labeled data is mandatory in the target domain, in order to induce the objective predictive function. The source and target labeled spaces can also be different.

- Transductive transfer learning: The target and source tasks are the same, but the domains are different. The target domain has no labeled data, and source and target feature spaces can be different.

- Unsupervised transfer learning: The target and source tasks are different, but related (explicit or implicit relationship between source and target domains). There is no labeled data in both domains, therefore this transfer learning category is applied in unsupervised learning problems, like clustering and dimensionality reduction.

Given the above definitions, only supervised learning can follow an inductive transfer learning approach since it is the only category that requires labeled data in both domains, which is always necessary in a supervised approach. Figure 2.1 schematizes the main goal of transfer learning.

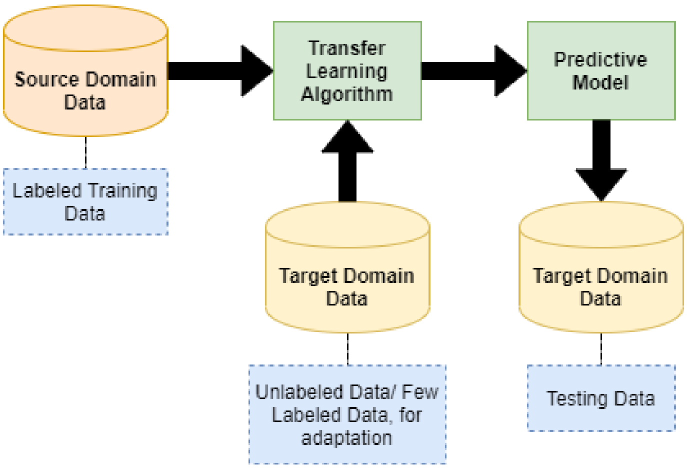


Figure 2.1: The goal of transfer learning (adapted from Pan and Yang (2010))

Regarding what type of knowledge can be transferred, four approaches can be distinguished:

- Instance-transfer: Follows the assumption that source and target domains have a lot of overlapping features, and manipulates the impact the source domain has on the target domain by re-weighting the first before re-using its instances in the target domain. For example, Dai et al. (2007) not only update the incorrectly classified examples in the target domain, but also does the same for source instances.

- Feature-representation-transfer: The source domain is used to learn a good feature representation that will be used in the target domain. As we will see later (in section 2.4.2.3), the BERT model resorts to this kind of transfer learning in its first unsupervised learning step (the pre-training), based on a large corpus, to capture patterns in language.

- Parameter-transfer: The motivation here is that if a well-trained source model has already learned various structures, since source and target tasks are related, those structures can be transferred to learn the target task. Thus, considering that the source and target models share some parameters or prior distributions of hyper-parameters, parameter-transfer focuses on regularizing those parameters in order to boost the target task's performance. Therefore, the shared parameter might assume different weights between source and target domains. For example, Gao et al. (2008) explored a dynamic locally weighted approach to combine multiple models as knowledge sources in the transfer learning process, managing the weights according to the impact each source model has on each test example in the target domain.

- Relational-knowledge-transfer: Assumes that there is some relationship among data in the source domain (e.g. connection between sentiment words and topic words), and that relationship is similar to the one existing in the target domain's data. Hence, in this approach we transfer the relationship among data from the source domain to the target domain. An example is the statistical relational learning approach of Mihalkova et al. (2007) that builds Markov Logic Networks (MLNs) which consider two domains that are related, and represent entities as predicates and their relationships as first-order logic. Then, between those two relational domains, we can find a mapping connecting entities and their relationships from the source to the target domains. For instance, the role played by a professor in an academic domain might be similar to the role played by a manager in an industrial management domain, and, consequently, the relationship between professor and students can be approximated to the one between manager and workers. Therefore, a MLN would present a link between professor and manager, and another link between the bound professor-student and the bound manager-worker.

Nevertheless, transfer learning is not always successful and we might incur to negative transfer, when the source domain data and task lead to a worse learning performance in the target task.

In NLP, transfer learning has been frequently used for Language model (LM) pre-training. Word representations are learnt from unlabeled data (Collobert and Weston, 2008), avoiding the need of extra annotation which can be time-consuming and expensive. Then, the obtained parameters can be used as a starting point for other supervised training models (Dai and Le, 2015). It has been shown that fine-tuning a pre-trained language model on a specific NLP task reveals successful improvements in the model performance on that task (Radford, 2018).

2.4 Text Representation

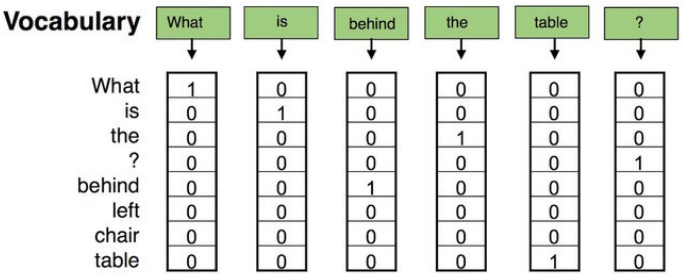
Data representation has a significant impact in machine learning performance. Machine learning models require a numeric representation of the words to perform calculations, so, good data transformations are needed in order to make it easier to extract useful information when building a classifier. In this section we talk about two types of text representation: **localist** and **distributed**. In the first, each entity is represented directly by a single and unique representation (one-to-one correspondence), while in the second, an entity is represented by a pattern of activation across a set of elements that can be shared over different representations (many-to-many relationship).

2.4.1 Localist Representation

> "In localist architectures each word within the lexicon is represented by a single unit." (Ralph, 1998)

Following the quote above, for the example of neural networks, each neuron is dedicated to only one concept. This representation is easy to understand and implement, but it can only represent a number of distinct concepts that is linear in the number of dimensions. This means that it cannot deal with the *curse of dimensionality*[1] because it requires $O(N)$ parameters to distinguish *O(N)* inputs (Bengio et al., 2013).

An example of this type of representation is **one-hot encoding**, where the feature representation is defined by the activation of a single element/position in a one-hot vector. One-hot vectors are used to represent words. One vector has the same size as the vocabulary and each vector only has one index with the value 1, corresponding to the respective word index in the vocabulary, and the rest of the values are 0. Therefore, a one-hot vector of length N can only represent N distinct values. Figure 2.2 shows an example of how a one-hot encoding model works.



Vocabulary	What	is	behind	the	table	?
What	1	0	0	0	0	0
is	0	1	0	0	0	0
the	0	0	0	1	0	0
?	0	0	0	0	0	1
behind	0	0	1	0	0	0
left	0	0	0	0	0	0
chair	0	0	0	0	0	0
table	0	0	0	0	1	0

Figure 2.2: Example of one-hot representation (extracted from Malinowski and Fritz (2016))

In light of the above, we can identify three problems in one-hot representations: data sparsity, incapability of handling out-of-vocabulary words, and jeopardising the relationship between different words by ignoring their context during word representation.

2.4.2 Distributed Representation

Distributed representation, again, considering neural networks, represents a concept by a pattern of activity over a set of neurons (Roy, 2017). This type of representation can describe a number of concepts that is exponential in the number of dimensions since more than one neuron is required to represent a concept and each neuron can be part of the representation of various concepts. Therefore, a representation vector of length N can express 2^N different values.

[1] When adding more dimensions brings exponential growth in data, and increases of unnecessary data, leading to insignificant value gains compared to the overhead.

Besides data sparsity (use of unnecessary vector space, with no valuable information), distributed representation addresses localist representation's inability to capture word semantics (as words are expressed independent of any context). It proposes an approach of encoding word meanings in their representations, allowing to directly understand the similarity between words. We are talking about **word embeddings**.

Word embedding is a set techniques to map words into a vector space. Since in word embedding each dimension represents a latent feature, it is expectable to find similar words distributed close to one another in the embedding space (words that have the same meaning have a similar representation). Hence, word embeddings can boost the performance of NLP tasks as it can capture context of a word in a text, semantic and syntactic similarity (by calculating the cosine similarity between vectors), and the relation between words.

The next two sub-sections present the two existing types of word embeddings. The first can be seen as static techniques, since a word will always be represented the same way, regardless of the context it is in. In contrast, the other type can be called as dynamic techniques and addresses the previous word embeddings' inability to capture polysemy by taking into consideration the context of the word.

2.4.2.1 Classic Word-Embeddings (context-free)

Mikolov et al. (2013) introduced **Word2Vec**, the first popular embeddings method for NLP tasks. It is a neural network with a single hidden layer (with the embeddings being its weights) and has two distinguished algorithms: **Continuous Bag Of Words** (CBOW) and **Skip-gram**.

The CBOW model uses the context (surrounding words) to predict a target word. The model input is an one-hot vector, representing the context. There are two weight matrices, one mapping the input to the hidden layer and another mapping the hidden layer outputs to the last output layer. The hidden layer simply copies the weighted sum of inputs to the next layer, resulting in a vector which elements are the softmax[2] values. The size of the output vector is the same as the size of the input vector. Figure 2.3 shows the architecture of a CBOW model.

On the other hand, the Skip-gram model uses a word to predict a target context. It takes the center word as input, and, considering a window of neighbor words (the context), tries to predict the context words by maximizing the probability of a word appearing in the context, given the center word. Similarly to the CBOW model, the input, center word, is represented by a one-hot vector. Figure 2.4 shows the architecture of a Skip-gram model.

Pennington et al. (2014) proposes **GloVe**, short for “Global Vectors", in order to overcome the disadvantages of context window-based methods, like Word2Vec Skip-gram, of not considering the total corpus statistics, not learning repetitions and large-scale patterns. GloVe is a count model, looking at how frequently a word appears in another word context (co-occurrence probabilities), within the whole corpus. Thus, this model learns the word representation vector by performing dimensionality reduction on a co-occurrence counts matrix.

[2] Neural network's final layer that yields probability scores for each class label. The softmax function takes a vector of K real numbers as input and normalizes it into a probability distribution, K probabilities.

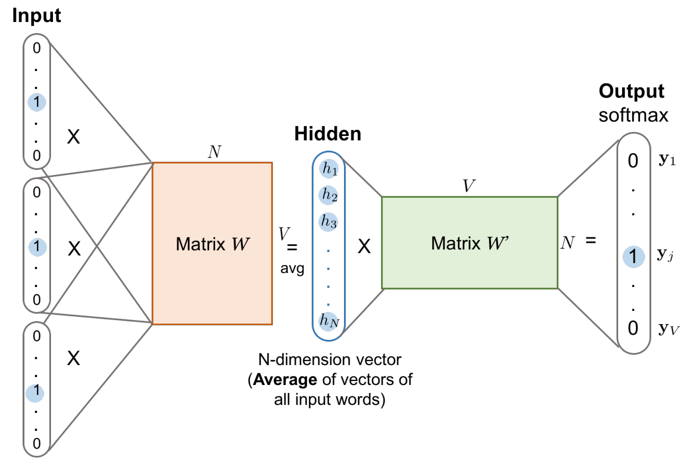


Figure 2.3: CBOW model (extracted from Lil'Log). N is equal to the vocabulary size and X refers to a word's one-hot encoding representation.

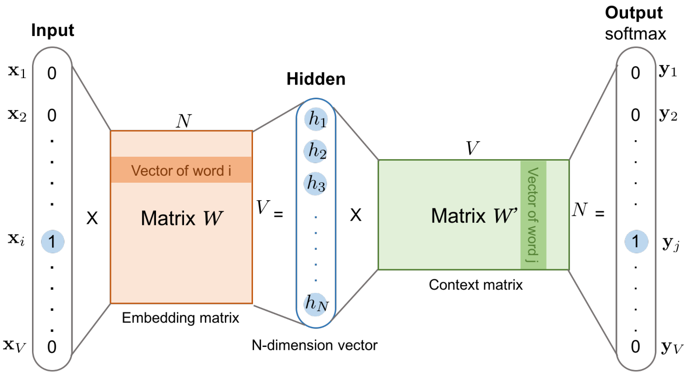


Figure 2.4: The skip-gram model. (extracted from Lil'Log). N is equal to the vocabulary size and X refers to a word's one-hot encoding representation.

fastText (Bojanowski et al., 2017) addresses Word2Vec's and GloVe's inability of handling out-of-vocabulary words. This model is based on the Skip-gram model, but, instead of using a distinct vector representation for each word which ignores its internal structure, each word is represented as a bag of character n-grams. Then, each character n-gram owns a vector representation, so a word is represented by the sum of its character n-grams' vector representations. As fastText model promotes different words to share same character n-gram representations, it is possible to learn the representation of rare words and of words that did not appear in the training data set.

To sum up, the techniques referred above generally produce a matrix used for lookup operations that map a word to a vector which is later fed into a neural network. The main issue with these models is that they generate the same word vector representation in different contexts.

2.4.2.2 Contextualised Word-Embeddings

Contextualised word embeddings tackle the problem of polysemy by capturing word semantics in various contexts. These methods rely on some language model, to help modeling word representations. **Language models** calculate the probability of a word being the next word in the given sequence, considering the previous words of that sequence. **Long short-term memory** (LSTM) models became very popular in language modelling at the time when they were first used for that purpose (Sundermeyer et al., 2012). An LSTM is an artificial recurrent neural network[3] (RNN), so, unlike feedforward neural networks, they have what is called feedback connections, and can process entire sequences of data.

Peters et al. (2018) proposed **Embeddings from Language Models** (ELMo) that generate vectors from a bidirectional LSTM[4], pretrained on a large corpus. The word representations are functions of the entire input sentence, learned from the internal hidden states of the deep bidirectional language model (biLM). The internal states are considered by computing a linear combination of the vectors yielded at the end of each internal layer, which proved to be superior to only using the top LSTM layer. The result is very rich word representations that incorporate both context-dependent aspects of word meaning (semantics), captured in higher-level LSTM states, and aspects of syntax, extracted in lower-level LSTM states. Moreover, ELMo can be easily integrated into existing models. Figure 2.5 depicts ELMo's architecture.

Vaswani et al. (2017) developed a model called **Transformer** that revealed to outperform LSTMs regarding dealing with long-term dependencies. Transformer is a model that relies entirely on **multi-headed self-attention**, replacing the RNNs frequently used in encoder-decoder architectures. **Attention** leads the model to focus on the relevant parts of the input sequence, hence, this technique aims to deal with the problem of having long sentences as inputs (Bahdanau et al., 2014) and speeds up the representation process. Self-attention is an attention mechanism that in order to generate the representation of a word, relates this word with other positions of the input sequence it is in. In this process, for each word embedding three vectors are generated – query, key and value vectors – from the multiplication of the embedding by three matrices (also called query, key and value matrices). Then, attentions are calculated through operations between the three vectors of different words in the input sequence. As referred at the beginning of this paragraph, Transformer, however, has a multi-headed attention mechanism which improves the model ability to focus on various positions and considers more than one "representation subspaces". For this mechanism, the only difference is that we have more than one attention head, and each of them starts with different query, key and value matrices, leading to distinguish attentions among those heads. At the end, the result matrices, obtained in each attention head, are combined into a single one.

Furthermore, the Transformer model is based on a multi encoder-decoder structure. Each encoder receives the input sequence or a sequence of continuous representations generated by its preceding encoder, applies self-attention, and passes its results, through a feed-forward network, to the next encoder or to the decoders (considering the last encoder in the stack of encoders). Then,

[3] Artificial neural networks that form a directed graph, along a temporal sequence. They use an internal state (memory) to process sequences of inputs.

[4] A bidirectional recurrent neural network offers a look-ahead ability (output layer getting information from both backward and forward states), by introducing, in each state, a new hidden layer that passes information to its previous state (information flowing in a backward direction).

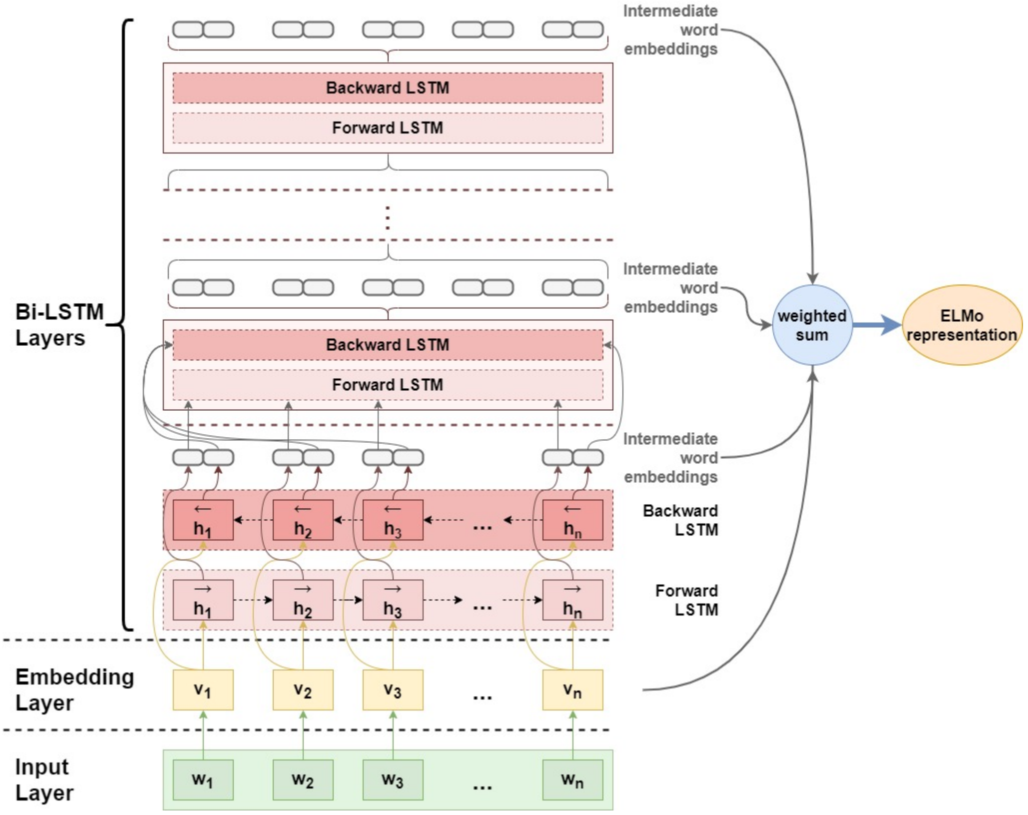


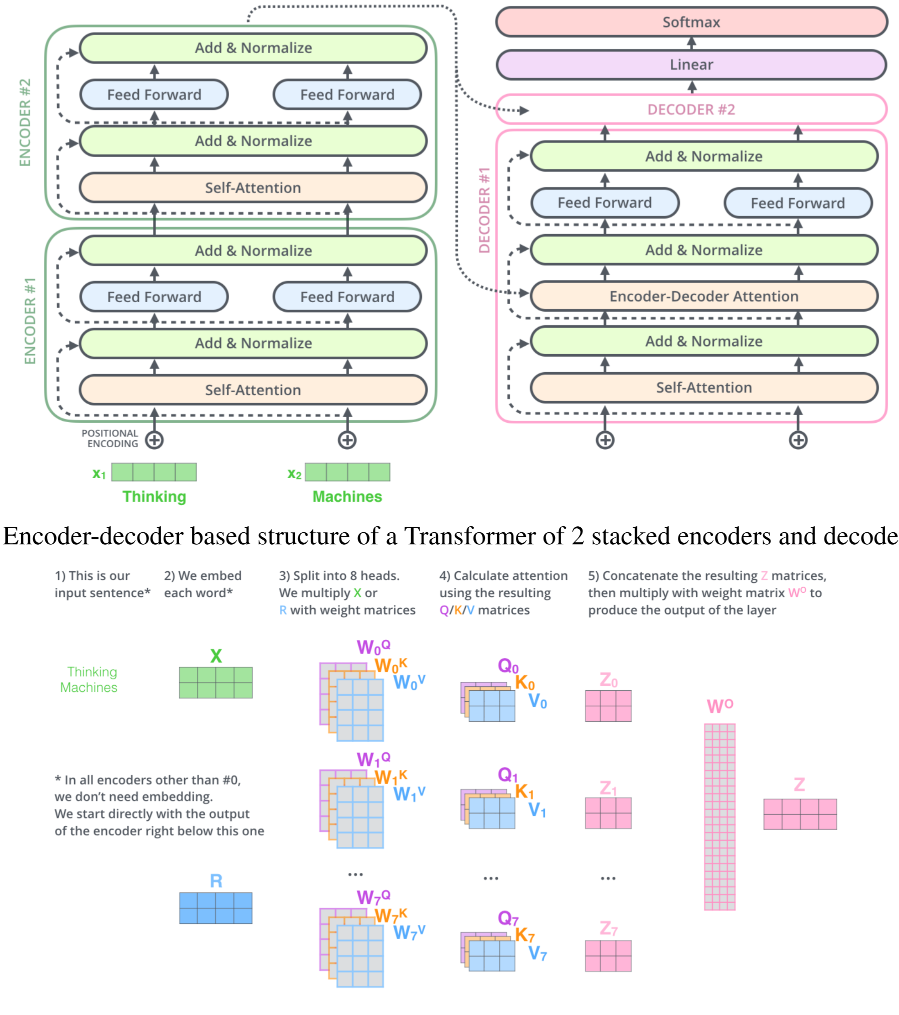
Figure 2.5: ELMo architecture (based on Yu et al. (2018), and A Step-by-Step NLP Guide to Learn ELMo for Extracting Features from Text, by Prateek Joshi).

a decoder generates, element by element, an output sequence of symbols of the same type as the ones in the input sequence.

Figure 2.6 depicts a summary of the Transformer model architecture and its multi-headed attention mechanism. The example input sequence shown is “Thinking Machines". The figure also presents two other layers, “Linear" and “Softmax". The linear layer is a fully connected neural network that maps the vector of floats, outputted by the stack of decoders, into a vector of the size of the model vocabulary, being each position the score of a single word. This linear layer is followed by a softmax layer that converts the words’ scores into probabilities.

Transformer showed to be a better approach for machine translation, since results obtained in learning dependencies between words through attention mechanisms are superior to the ones obtain with RNNs and their stateful nature. But, by itself, this model is not enough for sentence classification. For that purpose, Devlin et al. (2018) created the **Bidirectional Encoder Representations from Transformers** (BERT) model that, essentially, is a pre-trained Transformer Encoder stack which can then be fine-tuned for other downstream tasks[5]. Moreover, like ELMo, a pre-trained BERT can also be used to create contextualized word embeddings that are then fed to an existing model.

[5] Supervised-learning tasks that combine a pre-trained language model with a classifier.


ENCODER #2
ENCODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Self-Attention
POSITIONAL ENCODING
x_1
Thinking
x_2
Machines
Softmax
Linear
DECODER #2
DECODER #1
Encoder-Decoder Attention
1) This is our input sentence*
2) We embed each word*
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer
Thinking Machines
X
R
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
W_0^Q W_0^K W_0^V
W_1^Q W_1^K W_1^V
W_7^Q W_7^K W_7^V
Q_0 K_0 V_0
Q_1 K_1 V_1
Q_7 K_7 V_7
Z_0
Z_1
Z_7
...
W^O
Z

(a) Encoder-decoder based structure of a Transformer of 2 stacked encoders and decoders.

(b) Multi-headed self-attention mechanism, considering the example input sequence “Thinking Machines".

Figure 2.6: Overview of the Transformer model architecture (extracted from Jay Alammar’s blog The Illustrated Transformer).

2.4.2.3 Bidirectional Encoder Representations from Transformers (BERT)

BERT is an open-source[6] pre-trained language representation model, develop by Google (Devlin et al., 2018). In contrast to previous models that process words one-by-one, BERT performs a deep bidirectional representation by considering both left and right contexts. The model can be fine-tuned with just one additional output layer and minimal modifications.

As said before, BERT is based on a Transformer encoder. However, that transformer has different configurations, depending on the BERT model size (base or large), being larger than the Transformer default configuration presented by Vaswani et al. (2017) (which was 6 encoder layers, 512 hidden units, and 8 attention heads). For instance, the Base version has twelve encoder layers, feedforward networks of 768 hidden units, and twelve attention heads.

[6] https://github.com/ google-research/bert

Moreover, BERT enables parallelization in the training process because it is deeply bidirectional, in contrast to ELMo shallow bidirectional approach that generates the input representations (features for downstream tasks) through the concatenation of independently trained left-to-right and right-to-left LSTMs.

Figure 2.7 presents BERT architecture, depicting an overview of both pre-training and fine-tuning phases.

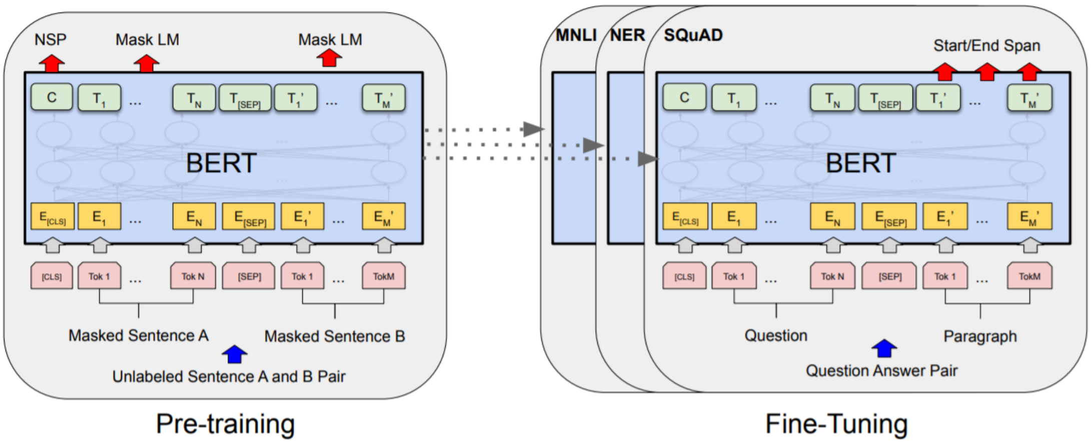


Figure 2.7: BERT architecture (figure extracted from Devlin et al. (2018)).

Tokenization

BERT conducts tokenization[7] using wordpieces (e.g. the word “playing" is divided into two subwords “play" and “##ing") instead of words. This allows resorting to a smaller vocabulary, and increases the useful information available for each word. Therefore, it improves the handling of out-of-vocabulary words. Still, if BERT fails to convert a word to wordspieces, using the available WordPiece vocabulary, then the word is represented by the special unknown token, [UNK].

Input/Output Representation

BERT can receive, as input, more than one sentence, by packing them together into one token sequence. That sequence always starts with the special classification token, [CLS]. The sentences, in the token sequence, are separated by a special token [SEP]. For each token, a vector is generated, however, for the classification, only the vector of the [CLS] token is used.

Pre-training

This step is conducted on a large corpus so the model can capture patterns in language. The

[7] The process of splitting a string into smaller tokens and, perhaps simultaneously, ignoring certain characters, such as punctuation.

corpus consists of 800M words from BooksCorpus and 2,500M words from English Wikipedia (only text passages), and, to get long contiguous sequences, BERT uses a document-level corpus. The pre-training is based on two unsupervised tasks:

- Masked LM: To achieve bidirectional representations, BERT follows a “masked language model” (MLM) pre-training. 15% of the sequence tokens are chosen randomly and they will be replaced with the mask token [MASK] or with a random token, with the respective probabilities of 80% and 10%. 10% of the time, the tokens are not changed. Next, the model tries to predict the original vocabulary word index of each selected token (deduce the missing words), based on the fused contexts from the left and right (using the final hidden vector representation of each token).

- Next Sentence Prediction (NSP): To have a model capable of understanding the relation between sentences, BERT does a “next sentence prediction" which is a joint pre-training of text-pair representations. Given two sentences, it predicts how likely is one to follow the other. The final hidden vector representation of [CLS] is used for this task.

Fine-tuning

Fine-tuning is a supervised training on a specific task with a labeled dataset. The task-specific inputs (features) are fed into the BERT model that generates the input tokens representations. The representations are used, along with the task-specific outputs (labels), to train the classifier (a softmax output layer which produces a probability distribution). The output layer (classifier) can receive all token representations or just the [CLS] representation, if the NLP task is, respectively, a token-level task (like tagging) or a classification problem (like entailment).

Furthermore, for sentence-pair tasks, like textual entailment and semantic similarity, fine-tuning takes advantage of the Transformer’s self-attention mechanism to include bidirectional cross-attention between two sentences, by encoding the concatenation of both.

SuperGLUE benchmark and state of the art results

The GLUE[8] benchmark is a evaluation framework for research in the field of natural language understanding (NLU). It also provides a ranking, based on a single-number metric, for model performance on different NLU tasks, built on existing public datasets. Since the recent developed models’ performances on this benchmark showed outstanding results, close to the level of non-expert humans, and even surpassing human performance on the same tasks (by 1.3 points in early July 2019), this benchmark version is, thus, unsuitable for measuring model performance and tracking their progress. Therefore, SuperGLUE[9] was introduced, presenting a new set of harder

[8] https://gluebenchmark.com/

[9] https://super.gluebenchmark.com/

Table 2.2: SuperGLUE Tasks.

Name	Identifier	Metric
Broadcoverage Diagnostics	AX-b	Matthew's Corr
CommitmentBank	CB	Avg. F1 / Accuracy
Choice of Plausible Alternatives	COPA	Accuracy
Multi-Sentence Reading Comprehension	MultiRC	F1a / EM
Recognizing Textual Entailment	RTE	Accuracy
Words in Context	WiC	Accuracy
The Winograd Schema Challenge	WSC	Accuracy
BoolQ	BoolQ	Accuracy
Reading Comprehension with Commonsense Reasoning	ReCoRD	F1 / Accuracy
Winogender Schema Diagnostics	AX-g	Gender Parity / Accuracy

NLU tasks, better resources, and a new public leaderboard. Moreover, it includes human performance estimates, for all benchmark tasks. Table 2.2 lists SuperGLUE current available tasks and the correspondent used metric to evaluate a model's performance in each task.

According to the public leaderboard[10], consulted on June 2020, SuperGLUE current state of the art score is achieved by T5 (Text-To-Text Transfer Transformer)[11], followed by variants of BERT. Hence, the best registered results in NLU tasks resort to Transformers, and are built around the BERT model.

2.5 Contradiction Detection

Contradiction is a semantic relation where two sentences cannot be true simultaneously. For instance, the two sentences “Some people and vehicles are on a crowded street" and “Some people and vehicles are on an empty street" are contradictory, as they claim two opposite ideas, the street being crowded or empty (Li et al., 2017).

In contrast to the above definition, Marneffe et al. (2008) consider that the appropriate definition of contradiction for NLP tasks, which allows to capture incompatibility between descriptions

[10] https://super.gluebenchmark.com/leaderboard

[11] https://github.com/google-research/text-to-text-transfer-transformer

of the same event, is that contradictions occur when two sentences are extremely unlikely to be true simultaneously. Besides annotating Recognizing Textual Entailment (RTE) datasets for contradiction, they then proposed a system where texts are represented as typed dependency graphs produced by Standford parsers, and contradictions are captured based on mismatches between the aligned texts graphs. For this task, they take into account seven features that reflect patterns of contradiction: polarity difference (words being negated, or not, by a negation dependency in the graph or by a linguistic marker of negation, such as “not", “no", and “few"); mismatches between numbers, dates, and times; antonyms (antonyms and contrasting words from WordNet (Miller, 1995), and oppositional verbs from VerbOcean (Chklovski and Pantel, 2004)); syntactic structures (the subject of one sentence overlapping the object of the other, and vice-versa); use of factive words; patterns of modal reasoning (based on the presence of modality markers such as "can" and "maybe"); and the relation between text elements. The results showed a lack of feature generalization, particularly in cases of contradictions marked by lexical and world knowledge. Harabagiu et al. (2006) took a similar approach of focusing on linguistic information (negation, antonymy, and semantic and pragmatic information), and addressing the recognition of contradictions, between two text inputs, as a classification problem that operates on the result of textual alignment (lexical alignment and paraphrase acquisition components).

Lin et al. (2003) created monothematic pairs, from RTE pairs, to highlight and isolate the linguistic phenomena (lexical, lexical-syntactic, syntactic, discourse, and reasoning phenomena) that more commonly give rise to contradictions and that contribute to the entailment relation. They concluded that the prominent phenomena in contradiction are quantity mismatching, semantic opposition (antonymy), mismatching oppositions, and general inference.

Kloetzer et al. (2013) tackled the most complex linguistic phenomena, the semantic relation, by proposing a method, at a text fragment level, for recognizing pairs of contradictory lexico-syntatic binary patterns (e.g., <"X promotes Y", "X prevents Y">). The method explores the interaction between contradiction and entailment by using three supervised classifiers, one for detecting the contradictory binary patterns, other to recognize entailment, and a last one trained with the entailment pairs and contradictions capture by the other two classifiers.

Regarding the issue of linguistic phenomena that require background knowledge (meronyms, synonyms, hypernyms, and reference ambiguity), Ritter et al. (2008), in order to verify whether phrases that initially appear contradictory are actually consistent statements, proposed an approach based on functional relations (relations accepting unique values of their arguments, mapping one entity to another and single entity). They converted sentences into tuples that represent the relation between a subject and an object, and use those tuples to find contradictory assertions. Shih et al. (2012) also faced the lack of background knowledge (e.g., limited number of antonyms available, and unstated common sense knowledge), but for a Chinese dataset. Therefore, they resort to the Web by first preparing mismatched conjunction phrases (queries), which consist of mismatches in sentence pairs, and then checking the number of hits in a Web search. They assume that implicit incompatibilities between two sentences originate a query that will have no or a low number of search hits.

As seen above, most of the studies on contradiction detection focus on investigating more explicit linguistic phenomena. Thus, they rely on a set of features for capturing patterns of contradiction, however, these features lack generalization. Still keeping in mind the linguistic phenomena, Dragos (2017) introduced uncertainty assessments. Uncertainty can be expressed when authors provide factual information with clues of how strong they support the reported facts, such as words expressing beliefs (e.g., "I believe", "I assume", "it seems"), lexical clues (e.g., "possibly", "probably", and "it is unlikely"), modal verbs, passive active language, and hedges (words that modify the uncertainty assigned to propositions). They consider sentences conveying factual information, represented through functional relations, and some degree of uncertainty. Then, for detecting contradictions, they resort to two relations between sentences, *disagreement* and *conflict*. Disagreement is conceived as two sentences, of similar semantic content on a shared topic, expressing different certainty of facts. Whereas, a conflict relation occurs when two sentences have opposite content on a shared topic, but similar uncertainty assessments. The opposite content arises at a lexical level, through negation, antonymy, and numerical mismatches, or from world knowledge.

On the other hand, Tsytsarau (2011), instead of following linguistic analysis and textual entailment, came up with an approach based on statistical principles and sentiment information. Hence, they proposed to detect time intervals where contradictions occurred, regarding some topic, at a large scale over time, based on the distribution of opposite sentiments. The sentiment towards a topic is a real number in the range [-1, 1] that represents the polarity and strength of the author's opinion. To identify contradictions, for each topic the sentiment values of different texts on that topic are gathered, and the sentiment average (the aggregated sentiment) and variance is calculated. They assume that an aggregated sentiment value close to zero and a high variance indicates a very contradictive topic. The data analysed was drug reviews, and comments on YouTube videos and on online short story posts from Slashdot. For each topic they store a time-tree structure where a node corresponds to a time window, summarizing information for all documents belonging to that interval, which allows to incrementally update the contradiction values over time. Likewise, Badache et al. (2018) explored reviews, related to a web resource, and sentiment analysis. They assumed that conflict of opinions about a specific aspect is followed by diversity of sentiments. So, they detect contradictions and calculate their intensity based on the sentiment polarity around the aspect, and on the rating associate with the online review.

Another strategy for contradiction detection is learning the semantics from the input by using word embeddings. However, context-free word embeddings, such as word2vec and GloVe, are not viable for this task since words with similar context will be mapped close to each other in the vector space, even if they have contrasting meanings (Devlin and Chang, 2018). Figure 2.8 shows such problem, considering the previous example referred in the first paragraph of this section.

In order to tackle this issue, Mrkšić et al. (2016), Chen et al. (2015) and Liu et al. (2015) benefited from public lexical databases, like WordNet (Fellbaum, 1998) and The Paraphrase Database (PPDB) (Ganitkevitch et al., 2013), to create semantic constraints which would reflect a better similarity relation between word vector representations. Still, these lexical resources are limited and

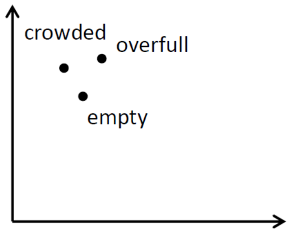


Figure 2.8: Traditional context-free word embedding mapping contrasting words into close vectors (extracted from Li et al. (2017)).

do not cover all existing antonym and synonym pairs. An improvement to these databases was the method proposed by Li et al. (2017) that can generate a bigger corpus of contrasting pairs which was then used to build a model that maximizes the semantic gap between contradictory words. Hence, they developed a feedforward neural network for learning contradiction-specific word embedding. Here, for representing the input words, they started by a trained embedding GloVe that is updated as the model is trained on the generated extensive corpus of contrasting pairs. Then, the learnt embeddings are used to represent the local and global semantic relations from the input sentences, serving as features for a Convolutional Neural Network model for contradiction detection. This approach outperforms the traditional context-free word embedding algorithms that map contrasting words into close vectors in an embedding space. Schwartz et al. (2015) toke a distinguish path by using symmetric patterns (SPs) to generate vectors representing two words that co-occur in a SP, allowing contradiction detection based on symmetric word relationships.

Table 2.3 presents some datasets that can be use for contradiction detection since the incorporate examples of document pairs labeled as contradiction.

2.6 Language patterns

Several researches were driven in understanding how different deceptive and truthful speeches are, through analysis of linguistic features (Jiang and Wilson, 2018), and how those language patterns can help improving prediction performances of whether a text content is true or not. Next, we will expose some of the various approaches conducted which explore the impact of linguistic cues in a model's performance.

Mihalcea and Strapparava (2009) studied patterns in word usage in deceptive and truthful texts using the words classes from Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2001) and calculating each class coverage (percentage of words from a text that are part of the class). They concluded that in deceptive texts the word classes from LIWC used more often are *metaphysic* (e.g. god, die, sacred), *human-related*, but detached from self (avoid self-involvement), and *certainty* (maybe for a more efficient persuasion).

Rubin and Vashchilko (2012) developed a methodology for deception detection based on Rhetorical Structure Theory (Mann and Thompson, 1987) analysis and applying a vector space

Table 2.3: Datasets containing examples of sentence pairs representing contradictions.

Dataset	Language	Date of publish	Short detail	Availability address
PHEME RTE (Lendvai et al., 2016)	English	2016	Tweets related to crisis events.	PHEME
Sentences Involving Compositional Knowledge (SICK) (Marelli et al., 2014)	English	2014	10,000 English sentence pairs annotated for relatedness in meaning (with relatedness scores) and entailment (being the possible labels “entailment", “contradiction", and “neutral").	SemEval-2014 Task 1
The Stanford Natural Language Inference (SNLI) (Bowman et al., 2015)	English	2015	570k human-written sentence pairs manually labelled as entailment, contradiction, and neutral.	The Stanford NLP Group
Multi-Genre Natural Language Inference (MultiNLI) (Williams et al., 2018)	English	2018	A collection of 433k sentence pairs annotated with textual entailment information, containing examples of different genres: “fiction" “government", “slate", “telephone", “travel", “9/11", “face-to-face", “letters", “oup", and “verbatim".	NYU

model (VSM) to represent RST relations (similarities in stories’ coherence and structure) in a story space, to later cluster them (one cluster for deceptive stories and another for truthful stories). They showed that RST dimensions can be used to distinguish between truthful and deceptive stories, as well as to identify different levels of deception.

Feng et al. (2012) investigated the syntactic style using, as model features, four different encodings of production rules based on Probabilistic Context Free Grammar (PCFG) parse trees, which proved to be helpful comparing with baselines (words represented by bag-of-words, and syntax and syntactic information encoded by part-of-speech tags).

Markowitz and Hancock (2014) investigated linguistic patterns in scientific reports, considering words related to causality, scientific methods and investigations, terms related to scientific reasoning, and language features used in describing scientific phenomena (quantities, terms expressing the degree of relative differences and words related to certainty). To analyze writing style, they used Wmatrix (Rayson, 2008), a tool that provides word frequency lists, and semantic and grammatical categories analyses. They found that fraudulent papers use more language to emphasize and relativize differences on findings, fewer words to soften or limit empirical findings and fewer adjectives.

Rashkin et al. (2017) explored linguistic attributes, through lexical resources such as LIWC, subjective words, lexicons for hedging and intensifying lexicons (words implying a degree of dramatization), to compare different types of fake news (propaganda, satire and hoax), by counting lexicons in tokenized texts. Additionally, they studied the feasibility of predicting news reliability (trusted, propaganda, satire or hoax), finding that the dominant features for each category were: specific places or times for trusted news; vaguely facetious hearsay for satire; divisive topics and dramatic cues for hoax; abstract generalities and specific issues for propaganda. They also concluded that the LIWC features do not improve the LSTM neural network used in their model for predicting truthfulness, maybe because some lexical information is redundant, as the model is capable enough of learning by itself that information directly from the text.

Volkova et al. (2017) developed a model to classify news as verified or suspicious using metadata (social graph and linguistic markers). It resorted to a large Twitter corpus, collected during the terrorist attacks in Brussels in 2016, that gathered tweets from both suspicious and verified news accounts. The sub-network that processes the text is an embedding layer followed by a LSTM layer or two 1-dimensional convolutional layers, and, finally, a max-pooling layer. The social graph is a feed-forward sub-network feed with one-hot vectors representing user interactions, whereas, linguistic markers can be bias, subjective, psycholinguistic and moral foundation cues. These two representations are incorporated through “late fusion" to train the network, which showed to have led to better prediction performance. The results show that suspicious news contain more bias markers, hedges and subjective terms, and moral cues.

Taking all the above into consideration, in deceptive texts we might be more prompt to face comparisons, lack of empirical results and details, like locations and time, and more abstract generalization, dramatization and subjectivity.

Even though, we are interested in detecting contradictions, these researches serve to consider that a target task, in this case detecting deceptive texts, can reveal characteristic language patterns, which can be helpful for improving an inference model performance on that task. Our target task is different from the one talked in this section, however it may also be marked by specific language patterns.

2.7 Summary

Natural Language Processing is the main scope of our project, still, in this section, besides listing some of the existing applications of NLP, we focus on contradiction detection. NLP is a very important and discussed area, however, there are not many studies regarding contradiction detection, leaving enough headroom for further research.

Indeed, when addressing NLP tasks, machine learning is the most popular approach. We have specially talked about supervised learning, giving emphasis to classification tasks. Hence, we came across requirements for building a classification model: the input representation, the output (multi-label or single-label classification, a discrete value or a continuous value), splitting the dataset for training and testing, inference method (numeric or symbolic), and performance evaluation.

In machine learning, a traditional method assumes that the training and testing data are in the same feature space and have the same distribution. Nevertheless, we not always have enough and perfect data, for learning and testing a task, thus we discuss transfer learning which tackles that issue by passing the knowledge learned in one domain to another domain, the one of interest. But this is not a silver bullet, and cannot be applied with whatever domain. There must be some relation between the source and the target domains in order to boost the model's learning performance. Otherwise we might face a negative transfer, having the opposite result, deteriorating the learning process.

Undoubtedly, data plays a crucial role in machine learning, and so does its representation. Localist representation was the first to be developed and uses a single unit to represent an item, leading to data sparsity, and lacking the representation of the relationship between items. Now researches and recent approaches resort to distributed representations, but localist is still used as an auxiliary in distributed architectures. In distributed representations, we can capture words semantics and the similarity between words by encoding them through word embeddings. Word embeddings can be context-free (static representations that do not deal with polysemy) or contextualised (the same word having different vector representations, depending on the context it appears). In fact, this last type of distributed representation is the most sophisticated way of encoding words, and so we present BERT, a recent model that combines Transformers and different attention mechanisms, proved to be the base of the state of the art approaches.

At last, we assess whether there are linguistic markers characteristic of deceiving texts, and, actually, previous researches, regarding language pattern, have proved so. Still, proving text's truthfulness is not the focus of our work, but this serves to present that language can reveal different behaviours and markers depending on its intent.

In this chapter we noticed that the NLP task of detecting contradictions is complex, and there are not sufficient studies around it. However, recent developments and trends in machine learning, such as the rapid evolution of contextualized text representations, and transfer learning (e.g., using pre-trained model, like BERT), might be key for new approaches regarding contradiction detection.

Chapter 3

Methodology

This chapter provides the description of the proposed methodology for the task of detecting contradictions, in a political domain as case study. Section 3.1 presents the concept of patterns in language, covering the political context and contradictions in texts. Section 3.2 details the proposed solution to address the problem of our task.

3.1 Language patterns and language in a political domain

Language is a powerful tool used for different objectives and might reflect those ambitions and author's personality. It can be manipulated in various ways, depending on the goal and context.

Jordan et al. (2019) found evidence, in multiple large corpora of American and other English-speaking elected leaders, of a decrease in formal (analytic) language, in contrast to an increase in informal and confident language, over time, in various political contexts. One of the factors that brought this trend is the evolution in communication technologies and mass media, as they come as both new opportunities and challenges for persuading the public and share messages.

Savoy (2018) specifically analysed the rhetoric and style adopted by Donald Trump and Hillary Clinton during the 2016 US presidential election, through different stylistic measurements, such as the most frequent used lemmas, sentence length (number of tokens), lexical density (informativeness of a text), frequency of big words, type-token ratio (ratio of rich vocabulary) and part-of-speech distribution. They also did a semantic-based analysis, by grouping words of same semantic meaning. They concluded that Trump follows a more direct style, using more verbs and adverbs, brief sentences consisting of short words, and repetitions, aiming to be understood by all in the audience. His style is associated to a strong masculine figure, energetic, nationalist and easy to understand. On the other hand, Clinton chooses a more descriptive rhetoric (more nouns, adjectives, prepositions, and determiners), using longer sentences with a richer vocabulary, covering more topics.

Finally, a characteristic relationship between language and politics can also be visible through the strategic use of personal pronouns and metaphors (Lin, 2012).

In light of the above, it is clear that there are characteristic behaviours in how we use language depending on the event and context we are in. Moreover, the target task also plays an important role in language. An example is Dragos (2017) that, when addressing contradiction detection, introduces the help of linguistic clues of uncertainty (the use of modal verbs, passive and active voice, “possibly", “probably", “might be", “it is unlikely", “undoubtedly", etc.).

3.2 Approach to detect contradictions in a political domain

In this work, we propose an approach to tackle the issue of detecting contradictions through a supervised classification model. As a case scenario, we choose to test in a political domain, to depict a specific relation between two documents, their topic similarity. Nevertheless, is important to highlight that our methodology does not intend to only draw conclusion for contradictions in politics. We rather aspire to explore whether we can take advantage of examples presenting different relations (e.g., the topic similarity, and arguments of attack) to infer the one we are testing, the relation of contradiction between two documents. We start by considering that language is context and goal driven. Thus, we believe that, through a supervised approach, datasets of a different, but similar, task (designed for another purpose) can additionally be used for learning our specific task of contradiction detection, in order to achieve better classification results. Therefore, this methodology's main aim is to study if those new datasets (source task domains) can be used to improve an inference model performance in predicting contradictions (target task domain). This process is called *Transfer Learning*. We also analyse if the source task domains by themselves (not using any examples of the target task domain for training a model) are enough to learn the target task.

To clarify our objective and supervised classification approach, formally we denote $D = \{D_1, ..., D_n\}$ as a size n collection of documents (text corpus). The model input is a pair of documents, $\langle C_1, C_2 \rangle$, with $C_1, C_2 \in D$. Considering $l \in \{0, 1\}$ as the possible label, where 0 and 1 represent, respectively, *not contradiction* and *contradiction*, the output will be the probability of the two documents being, or not, contradictory: $P(l | \langle C_1, C_2 \rangle)$.

Firstly, we define two assumptions that support the methodology:

Assumption 1 (The effect of the relation between documents in language)**.** *The difference/similarity in language used in two documents is (partly) determined by the relation between those documents.*

Assumption 2 (Patterns in contradictory statements)**.** *Language reveals particular patterns in contradictory statements.*

Here, in Assumption 1, by relationship between documents, we refer to all possible interactions and links, whether it is the context/topic they share, one text supporting the other, a relation of entailment, how similar they are, etc.

Regarding Assumption 2, as expressed before in section 3.1, contradictions in texts are identifiable by the language, independently of the domain, and, therefore, might share similar behaviours and patterns across various text genres. Furthermore, a contradiction does not necessarily need to be factual, where a text refers an event, date, location, or statistics different from established global conceptualizations and common knowledge. It can be a change in opinion. If a change in opinion is a conflict between two ideas proclaimed by a person or group in distinctive times, we can see a similar behaviour when a person disagrees with another person's allegation, since there is a collision in beliefs (a proposition that is incompatible with another proposition, or that don't make sense when presented together). Thus, we describe one hypothesis:

Hypothesis 1 (Different relations revealing similar effects)**.** *Different types of relationship between documents may reveal similar behaviours and effects on language.*

In the last paragraph we have talked about a possible relation that plays a similar role as contradictions. To better represent this hypothesis, we have the following example of illocutions[1] of disagreement, highlighted in bold, from the first Democratic primary debate of the 2016 United States presidential election held on October 13, 2015, in Las Vegas, Nevada (transcript available on CNN Press Room):

SANDERS: I think the governor gave a very good example about the weaknesses in that law and I think we have to take another look at it. But here is the point, Governor. **We can raise our voices, but I come from a rural state, and the views on gun control in rural states are different than in urban states, whether we like it or not.**
Our job is to bring people together around strong, commonsense gun legislation. I think there is a vast majority in this country who want to do the right thing, and I intend to lead the country in bringing our people together.

O'MALLEY: Senator — Senator, excuse me.

(CROSSTALK)

O'MALLEY: **Senator, it is not about rural — Senator, it was not about rural and urban.**

SANDERS: **It's exactly about rural.**

As mentioned in the previous chapter (Chapter 2, Section 2.5), Dragos (2017) considered, for the task of detecting contradictions, the degree of uncertainty expressed by lexical cues. In the above example of disagreement, we can also find such cues, the word "exactly" which marks how strong Sanders supports the reported fact.

We might also consider the two examples below extracted from Argument Annotated Essays corpus[2]. First, the case of a claim against the major claims, in a text. Second, a premise attacking a claim:

Example 1:

[1] *"an act performed by a speaker by virtue of uttering certain words, as for example the acts of promising or of threatening"* - Collins

[2] https://www.informatik.tu-darmstadt.de/ukp/research_6/data/argumentation_mining_1/argument_annotated_essays_version_2/index.en.jsp

Major Claim 1: “we should attach more importance to cooperation during primary education"

Major Claim 2: “a more cooperative attitudes towards life is more profitable in one’s success"

Claim against: “competition makes the society more effective"

Example 2:

Claim: “living and studying overseas is an irreplaceable experience when it comes to learn standing on your own feet"

Premise: “One who is living overseas will of course struggle with loneliness, living away from family and friends"

Here, it is hard for a human to detect patterns that can be also manifested in examples of contradiction. However, we choose the relation of attack as a possible source of knowledge to be transferred, due to its definition, of a claim against another, sharing similarities with the definition of contradictions. In both relations, one text questions the veracity and validity of another.

The three cases above are all examples of how we will explore the extension of other documents’ relations for our specific task of contradiction detection.

For this dissertation work, we have four main process phases, depicted in Figure A.1 from Appendix A.

First, defining the domain and two baseline datasets which are, in the context of transfer learning, our target task domain. After that, we analyse each baseline dataset by training and testing a neural network model with it. Then, for each new dataset (in the context of transfer learning, the source task domains), we train a neural network model with it and run predictions with examples from the two baseline datasets. Finally, we take each of the models trained in the previous step and retrain them with part of a baseline dataset and run predictions on the other part. We do this separately for both baselines.

Our methodology starts with the domain definition. Our test domain is indeed contradictions in a political context. For this domain, we have built two datasets. One includes only the examples of genre “government" from the Multi-Genre Natural Language Inference (MultiNLI) corpus[3]. The other is a set of manually collected and annotated pairs of documents, from different sources. Those documents are Donald Trump, the president of the United States, speeches in interviews and debates, posts in the social media network Twitter, and propositions from his published books. Some of those pairs are known to be contradictory. Because we want to explore how we can take advantage of information that was not initially designed for this particular scope, we distance a little from this domain and consider other datasets somehow related to the two previous ones. Hence, Table 3.1 shows the corpora used for that purpose.

As mentioned in Section 2.3, we can incur to negative transfer learning when the used source domain task leads to a worst learning performance in the target domain task. In this case, it is

[3] https://www.nyu.edu/projects/bowman/multinli/

better to discard that source domain task as it does not give valuable knowledge. We want to prove that if the learning performance increases, when using a new source dataset, it is due to the relationship between documents in those datatsets, and not because we are just increasing the number of training examples. In order to corroborate this idea, we also consider, for transfer learning, a dataset whose document-pair relationship is not related with the one presented in the two baseline datasets. In this situation, we expect a worst performance, sign of negative transfer learning. Otherwise, if the classification errors are too insignificant when using this dataset as source task domain, we assume that the main factor for transfer learning with the other source task domains was not necessarily the relationship between the documents, but perhaps just increasing the training set. This case would reveal that the new considered relations between documents are not relevant enough for transfer learning for the task of detecting contradictions. The last entry/row of Table 3.1 refers to the dataset used for this motive.

After gathering the required data, we train a set of binary classifiers using each dataset, and for a number e of epochs. Table 3.2 contains the class distribution for each dataset.

Considering each of the defined baseline datasets, we train and test a binary classifier with examples from the same dataset. Then, for each baseline dataset, we train a different binary classifier with that baseline dataset, and test with examples from another baseline dataset.

For the new datasets (MultiNLI-, US2016, ArgumentativeMicrotext, ArgumentEssays, and W2E), we train a binary classifier for each of them, and, separately, run predictions using each of the baseline datasets. Later, using these trained models, we perform retrainings, resorting to the baseline datasets. So, for one new dataset (source task domain) we consider two distinguish binary classifiers, both previously trained with the same source task domain data, but that will be retrained with different data, one from DonaldTrump's and the other from MultiNLIGovernment's.

After the experiments, we analyse the results through the study of the Receiver Operating Characteristic Curve (ROC curve), the Precision-Recall curve, the Area Under the Curve (AUC) of both curves, and the F1-score for different threshold values. We also examine the case where we consider as predicted label, for one input (pair of two documents), the one with higher score/probability value (threshold of 0.5).

This methodology aims to answer the two Research Questions already mentioned in Section 1.2 of Chapter 1:

Considering our target task of detecting whether two documents are contradictory or not, ...

$\mathscr{Q}_1$. *... can a classification model be effective when only trained with examples whose document-pair relations are different from the target one (contradictions)?*

$\mathscr{Q}_2$. *... can other examples, that incorporate document-pair relations different from the target one, be used to provide an extra training set of contradictory statements, in order to improve a model learning performance?*

For the first research question, we will see the performance of a model when trained on one non-baseline dataset and tested in one baseline dataset. Therefore, we train in one dataset (one

Table 3.1: The other datasets considered for the task.

Dataset	Language	Date of publish	Short-detail	Relation being tested	Availability address
Multi-Genre Natural Language Inference (MultiNLI)	English	2018	A collection of 433k sentence pairs annotated with textual entailment information, containing examples of different genres: “fiction", “government", “slate", “telephone", “travel", “9/11", “face-to-face", “letters", “oup", and “verbatim".	Contradiction in texts of different genres.	NYU
US2016	English	2019	Transcriptions of television debates leading up to the 2016 US presidential elections, and reactions to the debates on Reddit. The annotation of the corpus is based on Inference Anchoring Theory (IAT), containing three types of relations: inference, conflict and rephrase.	Arguments of disagreement between different speakers.	AIFdb
Argumentative Microtext Corpus	English and German	2015	Short texts that respond to a trigger question. The argumentation structure identifies the central claim of the text, supporting premises, possible objections and counters to these objections. The annotation guidelines are available online.	Author’s counter- -arguments attacking his\her own claims.	University of Potsdam
Argument Annotated Essays	English	2017	Argument annotated persuasive essays including annotations of argument components (“Major Claim", “Claim", and “Premise") and argumentative relations (“Support" and “Attack").	Author’s counter- -arguments attacking his\her own claims.	TU Darmstadt
Worldwide Event (W2E)	English	2018	Dataset for topic detection and tracking. 207,722 news articles covering a large set of 4,501 popular events, each belonging to one out of 10 categories.	Topic similarity.	W2E: A dataset for TDT

Table 3.2: Datasets dimension. The two first rows correspond to our two baseline datasets.

Dataset (our given name)	Short description	Total examples	Total positives	Total negatives
DonaldTrump	Manually created dataset, based on the article from POLITICO about moments where Donald Trump contradicts himself.	250	144	106
MultiNLIGovernment	All instances of genre “government" from the MultiNLI corpus.	79350	26418	52932
MultiNLI-	All instances, except the ones of “government" genre, from the MultiNLI corpus.	333352	110938	222414
US2016	US2016, the largest publicly available set of corpora of annotated dialogical argumentation.	1882	941	941
ArgumentativeMicrotext	The argumentative microtext corpus consists of short texts that respond to a trigger question.	1133	403	730
ArgumentEssays	Argument Annotated Essays corpus.	6673	715	5958
W2E	A Worldwide-Event Benchmark Dataset for TopicDetection and Tracking.	4800	2400	2400

source task domain) and run predictions with examples from a different dataset (one target task domain). If the model's performance is better than or close to the one when only using the baseline dataset for both training and testing, the new dataset used is reliable for learning the task. On the other hand, if it has a worse performance, but still good classification results, maybe the dataset is still good enough to be used for improving the learning of the task if later the model is refined with additional training, but now resorting to instances of the baseline dataset.

The second research question is answered by training a model with examples from a baseline dataset and a new dataset, test with different instances of the same baseline dataset used for training, and then verify if there were improvements in the classification results (through analyses of different evaluation metrics).

3.3 Summary

We provided an overview of how the text domain/topic can reveal particular behaviours in language, and the possibility that linguistic cues can reflect or help in detecting contradictions and deception.

We consider the wide range of the concept “contradiction" (not always just factual), speculating whether other relations between texts (e.g., disagreements, arguments of attack, premises against claims) can be seen as contradiction. Therefore, we propose a methodology to detect contradictions, exploring data not originally designed for this task. The approach consists of training a set of classifiers to predict if two documents are contradictory (1, positive class) or not (0, negative class). The performance of the models is used to conclude if the hypothesis of using information not directly targeted for contradictions, in a particular domain, can still be reliable for the task.

Chapter 4

Data and Experimental Setup

This chapter contains a description of the data used and of the experiments conducted to empirically test the methodology proposed in the previous chapter. Section 4.1 presents the various corpora details and how we take advantage of them to build our datasets. Section 4.2 describes the experimental procedures, driven in order to implement our methodology.

4.1 Datasets

4.1.1 Baselines

As said in the previous chapter, our work is focused on contradictions, and we will use the political domain as a case study. For that purpose, we use two datasets as baselines, one built by us from scratch, based on an online article, and the other containing a specific section of the publicly available corpus MultiNLI. Baseline datasets represent the target task domain in the context of transfer learning, therefore, the remaining datasets referred below (from Section 4.1.2 to Section 4.1.6) will be the source task domains.

4.1.1.1 DonaldTrump

For this dataset we focus on a specific entity, Donald Trump, the president of the United States, as a case study. The reason why we chose this well-known person is that there is a lot of controversy around his allegations, and the online magazine POLITICO Magazine [1] has an article exposing some of Trump's self-contradictions[2]. Hence, our domain is expected to be, mainly, political statements, although it can contain other topics escaping from our scope.

The article has a list of Trump's quotes in interviews, debates, posts in the social media network Twitter, and propositions from his published books. However, the list does not have a pattern, meaning that, you do not always have one quote followed by another that contradicts it. Thus, it requires to read and to analyse each quote. Moreover, not all the instances provide the source link

[1] https://www.politico.com/section/magazine

[2] https://www.politico.com/magazine/story/2016/05/donald-trump-2016-contradictions-213869

(e.g. when it is quotes from Trump's books), or the provided source link is sometimes unavailable or requires website subscription (like some blocked articles from The New York Times). So, in order to approve and verify quotes, we occasionally had to manually search on the internet for the quote, resorting to different sources, until we could find means to prove the reliability of the sentences.

Politico is an American political opinion company that produces contents covering politics and policy in the United States and internationally. It has professional journalists working for them to provide interesting, true and authentic content, so we assume we can trust this source. Nevertheless, some quotes are not easily identified as contradictions, as we can see in the following four examples:

Example 1:

> *"I love the poorly educated."*
>
> *"I see no value whatsoever in believing ignorance to be an attribute."*

We can argue that it is contradictory to be against ignorance, considering it an unacceptable "attribute", and, at the same time, claim to adore ignorant and poorly educated people. However, it is possible to judge the ignorance of people, but still like them.

Example 2:

> *"I'm very pro-choice."*
>
> *"And I am very, very proud to say that I am pro-life."*

Here it is impossible to capture the contradiction if you do not know the meaning of the concepts "pro-choice"[3] and "pro-life"[4].

Example 3:

> *"Everybody kisses your ass when you're hot. If you're not hot, they don't even call. So it's always good to stay hot."*
>
> *"He thinks he's hot stuff. And I hate people that think they're hot stuff, and they're nothing."*

While Donald Trump believes that it is good to "stay hot", he also says that he hates "people that think they're hot stuff". But, is he talking about everyone who thinks is "hot stuff" or the ones that think that, but, in fact, they are not? If we consider the first case, then it would not make sense supporting "to stay hot" and hate those who think they are. On the other hand, the second scenario does not create conflicts in recommending to "stay hot" and hating people that think they are more popular than they actually are.

[3] Favour the legal right of a woman to choose whether or not she will have an abortion.

[4] Opposing abortion and euthanasia.

Example 4:

> *“And I win, I win, I always win. In the end I always win, whether it’s in golf, whether it’s in tennis, whether it’s in life, I just always win. And I tell people I always win, because I do."*
>
> *“I want to win, and I’m not happy about not winning."*

If Trump always wins, it is contradictory to consider the case where he does not. Yet, the second quote might not be considering the possibility of losing, but rather highlighting how eager he is to win.

While verifying the used quotes, we could also find paraphrases or similar phrases, also said by Donald Trump, and used them to extend the dataset. For example, when searching for *“Here’s a man that not only got elected, I think he’s doing a really good job."*, we managed to find *“I think that he’s really doing a nice job in terms of representation of this country. And he represents such a large part of the country."* and *“Well, I really like him. I think that he’s working very hard."*. These similar instances were not only used to increase the positive examples, but also to generate negative examples, because a pair of equivalent texts cannot be contradictory. Therefore, if there was a pair of documents $\langle D_1, D_2 \rangle$ known to be contradictory, and later we would find a third document D_1' which meaning and content is similar (both documents expressing the same idea) to D_1, then we would generate a new positive example $\langle D_1', D_2 \rangle$ and a new negative example $\langle D_1, D_1' \rangle$.

Besides the negative examples formed from paraphrases that we found as we looked for evidence, we also resort to the platform Factbase that provides the entire corpus of Donald Trump’s public, and unedited, statements and recordings. The transcribed information is linked directly to the originating source. For this process, we filtered transcripts by keywords, like “gun control", or just opened random transcripts. Then, we extracted, from those selected transcripts, sentences where Trump would repeat the same idea.

To analyse the distribution of topics through positive and negative examples, we used the Latent Dirichlet Allocation (LDA) algorithm, for topic modeling, through python’s library Scikit learn.

Before generating the topics, we consider the unique instances of all dataset examples (we do not use duplicate documents), and perform data cleaning (remove backslashes, commas and semicolons) and tokenization.

Then, to create the document word matrix, the LDA model main input, we use CountVectorizer. We configured it to ignore terms that appear in more than 95% of the documents (max_df=0.95) and terms that appear in less than 2 documents (min_df=2), to remove built-in english stopwords (stop_words=‘english’), to convert all words to lowercase (lowercase=True), and to impose that a word has to contain numbers and/or alphabets, of at least length 3, in order to be qualified as a word (token_pattern=‘[a-zA-Z0-9]{3, }’).

When building the LDA model, we set the number of topics to 15 (n_components=15), the maximum learning iterations to 5 (max_iter=5), the learning method to online (learning_method=

Table 4.1: The number of instances of each topic that appear in positive examples (contradictions) and negative examples, and the difference in frequency of topic occurrence (Diff.) between these two classes.

Topics	Positive examples	Negative examples	Diff.
great	24	3	**21**
oil	14	**43**	**-29**
win	20	10	10
love	11	6	5
like	14	21	-7
people	31	13	**18**
dont	**39**	**39**	0
jobs	3	10	-7
think	**39**	19	**20**
pro	**41**	7	**34**
penalty	16	5	11
going	18	4	14
cancer	4	8	-4
thinker	8	2	6
years	6	**22**	-16

‘online’), the learning offset/tau_0 to 50 (learning_offset=50.), and the seed used by the random number generator to 0 (random_state=0).

Table 4.1 shows the obtained topics and the distribution of each topic for all positive examples (pair of two documents) and for all negative examples. The LDA model performance is out of the scope and its the denomination of each of the 15 obtained topics is not that relevant. We just want to explore the difference in frequency of topic occurrence (Diff.) between the two classes.

According to Table 4.1, the three most frequent topics in positive examples (pairs of contradictions) are “pro", “think" and “dont", whereas for the negative examples are “oil", “dont" and “years". The top five of most unbalanced topic distributions between the two classes are “pro", “oil", “great", “think" and “people". These values may have an impact on the experimental results, as we consider the possibility of having the model predicting based on the input topic, instead of predicting based on the relation of contradiction between two given documents.

It is important to remember that there might be other factors influencing and creating bias in this dataset, since it was not built by trained annotators.

4.1.1.2 MultiNLIGovernment

The Multi-Genre Natural Language Inference (MultiNLI)[5] corpus (Williams et al., 2018) addresses the coverage limitation faced by other Natural Language Inference (NLI) datasets, in terms of variety of meanings, expressed in English. It is one of the largest corpora for NLI tasks, and

[5] https://www.nyu.edu/projects/bowman/multinli/

includes ten distinct genres of written and spoken English. The wide range of styles, degrees of formality, and topics introduce greater linguistic difficulty and diversity, and make this corpus a benchmark for cross-genre domain adaptation. Moreover, the MultiNLI dataset allows to evaluate a model's ability to generate sentence representations in unfamiliar domains (cross-domain transfer learning). These characteristics and objectives meet our purpose too, as we want to mitigate whether we can take advantage of unknown and different, but still similar, document relations for our specific task of detecting contradictions.

Nine of the genres were extracted from the second release of the Open American National Corpus (OANC):

- **FACE-TO-FACE** genre uses transcriptions from the Charlotte Narrative and Conversation Collection of two-sided conversations.
- **GOVERNMENT** genre uses reports, speeches, letters, and press releases from public domain government websites.
- **LETTERS** genre uses letters from the Indiana Center for Intercultural Communication of Philanthropic Fundraising Discourse.
- **9/11** genre resorts to the public report from the National Commission on Terrorist Attacks Upon the United States.
- **OUP** genre uses five non-fiction works on the textile industry and child development, published by the Oxford University Press.
- **SLATE** genre uses popular culture articles from the archives of Slate Magazine.
- **TELEPHONE** genre uses transcriptions from University of Pennsylvania's Linguistic Data Consortium Switchboard corpus of two-sided telephone conversations.
- **TRAVEL** genre uses travel guides published by Berlitz Publishing.
- **VERBATIM** genre uses short posts about linguistics for non-specialists from the Verbatim archives.

The tenth genre, **FICTION**, uses several freely available works of contemporary fiction.

MultiNLIGovernment is the name we gave to the subset of MultiNLI corpus that only includes the examples of “government" genre. MultiNLI corpus has three possible labels:

- **Entailment**: relation between two sentences, a premise and a hypothesis, where the hypothesis is necessarily true or appropriate whenever the premise is true.
- **Contradiction**: relation between two sentences, a premise and a hypothesis, where the hypothesis is necessarily false or inappropriate whenever the premise is true.
- **Neutral**: relation between two sentences, a premise and a hypothesis, where none of the above conditions (entailment and contradiction) are applicable.

Since our task is to detect contradictions, using only two labels (0 and 1, respectively *not contradiction* and *contradiction*), we consider the labels “entailment" and “neutral" to be negative examples (not contradictions).

To built the dataset we used the JSON Lines format of the corpus. We filter the objects of ‘genre’ ‘government’ because, as said before, we are considering a political domain and the “government" type belongs to that field. To create an input pair, we use object’s values for ‘sentence1’, ‘sentence2’ and ‘gold_label’. Gold-label is the label used for classification. In the validation process of the MultiNLI corpus, when an example does not receive a three-vote consensus on any label, the golden-label is ‘-’. In this case, we consider it to be a negative example.

We use the python’s library Pandas to use its data structure DataFrame. We create three data frames, for test, validation and train sets. We split 70% of the obtained data for training, 10% for validation, and 20% for testing, then we save each set in a tab separated text format (‘test.tsv’, ‘dev.tsv’ and ‘train.tsv’).

At the end, we got a total of 79,350 examples, 26,418 positives and 52,932 negatives (Table 3.2).

4.1.2 MultiNLI-

This dataset follows the same procedure as the one described for the MultiNLIGovernment dataset. However, since we aim at exploring a model’s learning performance when giving data containing different relations from the ones we are using as baselines (contradictions in a political domain), we use all the examples of the MultiNLI corpus, except the ones of “government" genre. We also split the data in two sets, 80% for training (‘train.tsv’) and 20% for testing (‘test.tsv’). In this case, we got a total of 333,352 examples, 110,938 positives and 222,414 negatives (Table 3.2).

We decided to ignore the examples of “government" genre because we want to explore the behaviour of different document relations. Since we are going to use the political domain as a case study, we remove the “government" genre to only include genres that are not closely related to the political field.

4.1.3 US2016

US2016[6](Visser et al., 2019) is the largest corpus of annotated dialogical argumentation[7]. It comprises transcripts, collected from The American Presidency Project, of televised debates leading up to the 2016 presidential election in the United States of America: the first Republican primary debate on 6 August 2015 in Cleveland, Ohio; the first Democrat primary debate on 13 October 2015 in Las Vegas, Nevada; and the first general election debate between Hillary Clinton and Donald Trump on 26 September 2016 in Hempstead, New York. Therefore, the domain is argumentation in political debate. US2016 also includes online reactions, from Reddit, towards the three presented debates. Anyone who is a registered user in this social media platform can make posts,

[6] http://www.corpora.aifdb.org/US2016

[7] Argumentation is reasoning in discourse to support a contested point of view. To resolve disagreements, arguments can be used and the reason supporting them can be tested.

which leads to a greater diversity in language used, due to having people contributing from varying backgrounds, nationalities and education levels. Thus, for the online reactions, it is expected a mixed argumentative quality (rhetorical efficacy, and dialectical and logical fallaciousness) and many less well-crafted and well-signalled examples.

We are again facing cross-genre data, as we have both televised election debates and social media discussions. The US2016 corpus is a set of “argument maps" that are the result of the text annotation. It is organized in sub-corpora related to either the television debated transcripts (US2016tv) or Reddit threads (US2016reddit), for each of the three candidate debates preceding the 2016 US presidential elections (US2016R1, US2016D1 and US2016G1): US2016R1tv, US2016R1reddit, US2016D1tv, US2016D1reddit, US2016G1tv, and US2016G1reddit.

The data annotation format is based on Inference Anchoring Theory (IAT) (Budzynska and Reed, 2011). IAT adheres to the extended Argument Interchanged Format (AIF+) standard which is a graph-based ontology that facilitates the representation of arguments. For the annotation, we have the following concepts:

- **Location**: speaker identification followed by an argumentative discourse unit (ADU) which is a segmented transcribed text, that has a discrete argumentative function (right top and bottom boxes in Figure 4.1).

- **Transitions**: functional relation between locutions, representing the dialogue protocol (right middle box in Figure 4.1).

- **Illocutions**: the intended communicative function of a locution or of a transition between two locutions (middle column of boxes in Figure 4.1), and can be agreeing, arguing, asserting, challenging, disagreeing, questioning, restating and default illocution.

- **Proposition**: propositional content reconstructed from a locution (left top and bottom boxes in Figure 4.1).

- **Inference**: relation between two propositions where one supplies a reason for accepting the other (premise of an argument supporting its conclusion).

- **Conflict**: relation between two propositions where one is incompatible with the other (left middle box in Figure 4.1).

- **Rephrase**: relation between two propositions where one is meant to be a reformulation of another proposition.

Figure 4.1 shows an example of disagreement and how the argumentation is anchored in the structure of the dialogue. The blue boxes on the right are locutions and the ones on the left are the correspondent propositions.

Since we are proposing a binary classification model, we only use two labels (0 and 1, respectively, *not contradiction* and *contradiction*). In this dataset, we see both inference and rephrase

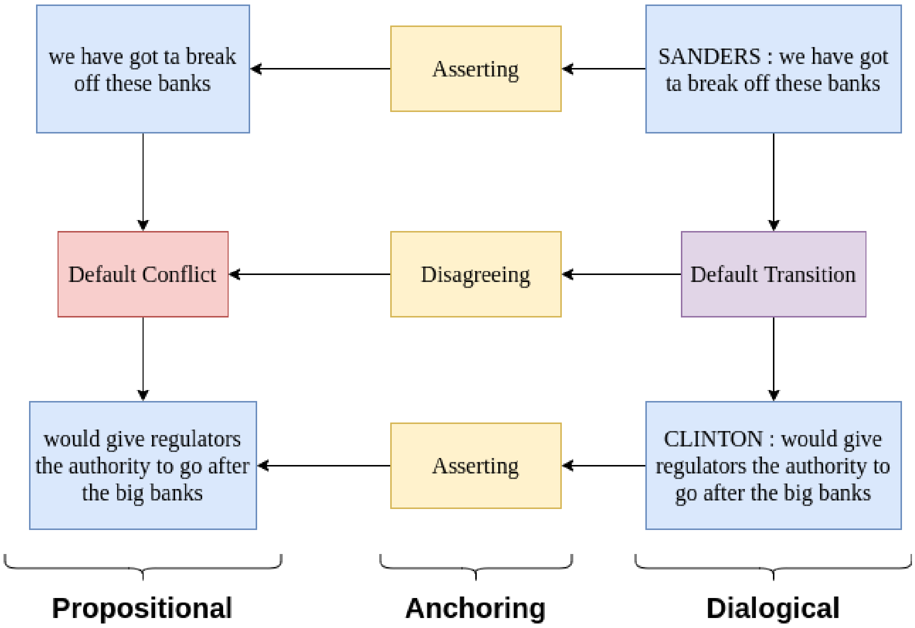

we have got ta break off these banks
Asserting
SANDERS : we have got ta break off these banks
Default Conflict
Disagreeing
Default Transition
would give regulators the authority to go after the big banks
Asserting
CLINTON : would give regulators the authority to go after the big banks
Propositional
Anchoring
Dialogical

Figure 4.1: Diagrammatic visualisation of an example of disagreement showing how the propositional reasoning on the left is anchored in the dialogical realisation of the argument on the right. This example was taken from US2016 corpus and is available online at http://www.aifdb.org/argview/10439.

relations as negative examples (not contradictions) and the relation of conflict as a positive example. A relation of conflict is linked to a disagreement illocution. A disagreement occurs when two interlocutors dispute the acceptability of a standpoint (an opinion, a belief, a proposal). They can, then, give arguments in order to resolve the disagreement while testing the reasons supporting their arguments. Hence, we are talking about the case when two people share different opinions which we see as a contradiction when considering both points of view as true.

To build our dataset we use the entire US2016 corpus in JSON format. The JSON includes a list of nodes and a list of edges. Each node is an object that has ‘nodeID’, ‘text’, ‘type’ and ‘timestamp’. Below we present the seven possible node types:

- **L -** Locutions, excerpts from the used transcripts. In this case, the node text is the extracted snippet.
- **TA -** A transitions (link between locutions). In this case, the node text is “Default Transition".
- **YA -** Illocutions that link locutions to propositions. In this case, the node text can be “Agreeing", “Arguing", “Asserting", “Challenging", “Default Illocuting", “Disagreeing", “Restating", and “Questioning".
- **I -** Proposition. In this case, the node text is a processed locution.
- **RA -** Relation of inference which is a link between two propositions where one gives a reason for the other to be accepted. In this case, the node text is “Default Inference".

- **CA -** Relation of conflict which is a link between two propositions where one is an incompatible alternative to another. In this case, the node text is “Default Conflict".
- **MA -** Relation of rephrase which is a link between two propositions where one reformulates the other. In this case, the node text is “Default Rephrase".

An edge represents the connection between two nodes. It has an ‘edgeID’, ‘fromID’ (id of the source node), ‘toID’ (id of the destination node), and ‘formID’ (which is always “null"). Therefore, to form an input pair, we get the ‘nodeID’ from a node of type CA. That node would be the red box from Figure 4.1. Then, we need two edges, one that has the CA node as ‘fromID’ and other that has it as ‘toID’. There will be two distinct edges with the CA node as ‘toID’ because, besides the proposition node (blue box in Figure 4.1 upper left corner), there is always an illocution node (yellow boxes in the middle of Figure 4.1) anchoring the propositional reasoning to the dialogical act (linking the boxes on the right side to the boxes on the left side, in Figure 4.1).

From the edges coming and leaving the CA node, we get the proposition nodes (type I). The text of those two nodes are the sentences of the input and, in this case, the label will be 1 (contradiction). For the negative examples (label 0), we follow the same procedure, but resorting to relation nodes of type RA and MA.

Regarding the count of propositional relations, this corpus has 2830 inference relations, 942 Conflict relations and 764 Rephrase relations. In our dataset we keep a balanced ratio of positive and negative examples by using 941 conflict relations (one of the conflict relations had an error since it was missing a node linking to the CA node) and getting a total of 941 examples from both inference and rephrase relations. We give priority to the examples of rephrase because in this relation it is more clear that the two sentences do not conflict, since one is basically paraphrasing the other. However, we ended up not using all the 764 rephrase instances because there were ones that were too small and simple to matter, like the following examples:

1. “CHINA. Mexico" and “Mexico. CHINA"
2. “flat tax" and “I ’ve advocated a proportional tax system"
3. “X for TRUMP’s family" and “X"
4. “Wrong. Wrong wrong" and “Wrong"

Thus, for the negative examples, we only consider pairs in which each sentence has at least four words.

Finally, we shuffle all obtained examples and split them in two sets, 80% for training (‘train.tsv’) and 20% for validation (‘dev.tsv’).

4.1.4 ArgumentativeMicrotext

The Argumentative Microtext Corpus[8] is the result of argumentation mining which involves capturing the different aspects of the argumentation structure of a text (central claim, supporting

[8] http://angcl.ling.uni-potsdam.de/resources/argmicro.html

reasons, possible objections, counters to the objections). Thus, the argumentation structure of a text is a graph representation, depicting the argumentative relation between the propositions.

The corpus provides short texts which are responses to trigger questions and is divided in two parts. The first part (Peldszus and Stede, 2016) has 122 texts: 89 texts collected in a controlled text generation experiment based on a list of controversial questions[9], and 23 texts written by Andreas Peldszus, as a "proof of concept" for the idea, and with the purpose of teaching and testing students argumentative analysis. The second part (Skeppstedt et al., 2018) was produced by a crowdsourcing experiment, also based on a list of trigger questions[10], resulting in 171 more texts.

The annotation scheme is based on the idea of modeling the argumentation as a hypothetical discussion between the proponent, who presents and defends its claims, and the opponent, who question and criticizes them. However, each microtext of this corpus only has one author that not only gives reasons in favour of the main claim, but may also take counter-arguments into consideration. Figure 4.2 shows the schematic diagram of one of the corpus microtexts. The nodes represent propositions extracted from text segments (the grey boxes). The shape of the nodes indicates the role of the correspondent proposition: round nodes are in favour of the claim and square nodes against it. The arrowhead, circle-head and square-head edges represent, respectively, a supporting move, an attacking move of rebuttal (challenging the acceptability of a proposition), and an attacking move of undercutter (challenging the acceptability of an inference between two propositions). In the example in Figure 4.2, the fourth segment rebuts the first segment, and this rebutting move is undercut by the fifth segment.

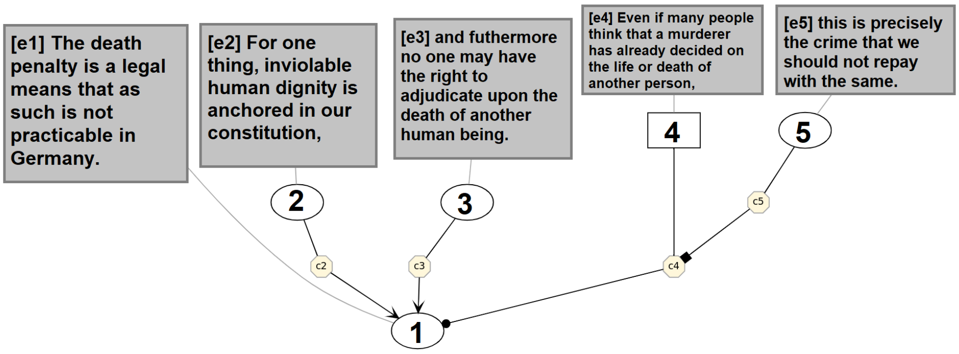

[e1] The death penalty is a legal means that as such is not practicable in Germany.
[e2] For one thing, inviolable human dignity is anchored in our constitution,
[e3] and futhermore no one may have the right to adjudicate upon the death of another human being.
[e4] Even if many people think that a murderer has already decided on the life or death of another person,
[e5] this is precisely the crime that we should not repay with the same.
1
2
3
4
5
c2
c3
c4
c5

Figure 4.2: Microtext and argumentation graph. This example was taken from the first part of the Argumentative Microtext Corpus and is available online at https://github.com/peldszus/arg-microtexts/blob/master/corpus/en/micro_b006.pdf

To build our dataset, we resort to the corpus XML format. The XML representing a microtext graph has elementary discourse unit (EDU) elements and argumentative discourse units (ADU) elements, which are EDUs that serve as independent arguments to the argumentation. The EDU

[9] https://github.com/peldszus/arg-microtexts/blob/master/topics_triggers.md
[10] https://github.com/discourse-lab/arg-microtexts-part2/blob/master/topics_triggers.md

element's content is character data (CDATA) presenting a text segment, and the ADU element has an attribute 'type' that says if the text segment supports ("type="pro"") or refutes/attacks ("type="opp"") the main claim. The XML also has edge elements with four attributes: "id", "src" (element from where the edge is leaving), "trg" (edge destination element), and "type" (type of link between two XML elements). We are interested in four edge types:

- **seg -** an edge of this type connects an EDU element to its correspondent ADU element.
- **sup -** an edge of this type represents a relation of support, connecting an ADU element to another ADU element, with the objective of increasing the credibility of the second ("trg" element) by providing a reason ("scr" element) for accepting it.
- **reb -** an edge of this type represents a rebutter (attack between propositions), connecting an ADU element to another ADU element, using the first ("src" element) to refute or weaken the force of the second ADU ("trg" element).
- **und -** an edge of this type represents an undercutter (attack to the relation between propositions), connecting an ADU element to an edge element, using the first ("src" element) to challenge the second ("trg" element).

In this scenario, we will use the support relations as negative examples and the attack relations as positive examples, because a supporting statement aims to increase the strength of the argument and a attacking statement aims to refute the target.

Regarding rebutters, if the two elements of this relation have other elements directly supporting them, we concatenate those text segments to give more context to the document used in the input pair. In the example from Figure 4.3, where the trigger question is "Should the statutory retirement age remain at 63 years in the future?", text segments two and three support the first, and the fourth text segment rebuts the first, so, here, the input pair with label 1 (contradiction), would be ⟨*"The implementation of retirement at 63 is no longer socially sustainable, as the population in Germany has, viewed demographically, a disproportionate number of old people, and constantly declining birth rates are being recorded." ; "Admittedly the number of immigrants is constantly rising in Germany"*⟩. The first document of this input pair talks about the struggle of retiring people at the age of 63 because the elderly population in Germany is increasing and the birth rate is decreasing. By doing so, Germany would lose many employees, ending up missing workers. The second document of the pair attacks the first by stating that the number of immigrants is rising which can help covering the lack of employees created due to the possibility of retiring at 63. Thus, the issue of the first document would no longer be a problem. That is why we see this as a contradiction, because if the counter argument holds, then the first claim loses its strength and/or meaning. Still, it is not truly a contradiction, since it is not certain that the fact expressed in the second statement is a solution for the problem presented in the first statement, hence both statements might co-exist (be simultaneously true).

Regarding undercutters, since the target element ("trg") is a relation (an edge) between two elements, from that relation we get the two linked elements/nodes. Then, we only use the ones

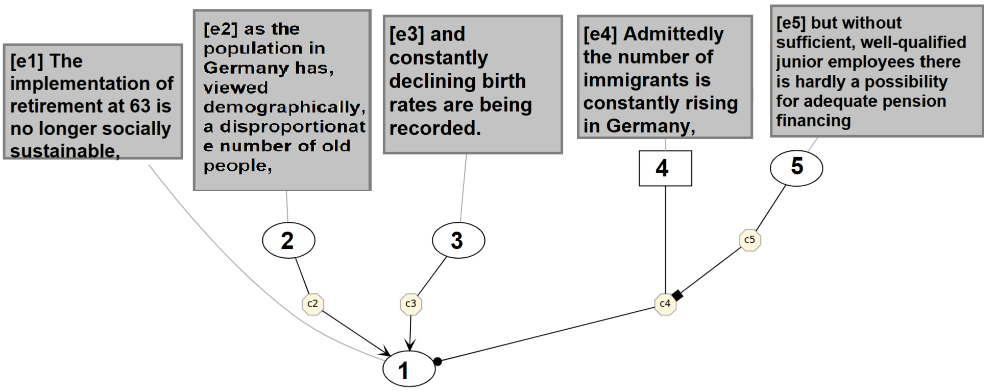

[e1] The implementation of retirement at 63 is no longer socially sustainable,
[e2] as the population in Germany has, viewed demographically, a disproportionate number of old people,
[e3] and constantly declining birth rates are being recorded.
[e4] Admittedly the number of immigrants is constantly rising in Germany,
[e5] but without sufficient, well-qualified junior employees there is hardly a possibility for adequate pension financing
1
2
3
4
5
c2
c3
c4
c5

Figure 4.3: Microtext and argumentation graph. This example was taken from the first part of the Argumentative Microtext Corpus and is available online at https://github.com/peldszus/arg-microtexts/blob/master/corpus/en/micro_k017.pdf

that are of a different type ("pro" or "opp") as compared to the undecutter's source element ("src"). In the example from Figure 4.3, the fifth text segment undercuts the relation of attack between segments one and four. Since the fourth text segment is of proponent type ("type="pro"") ant the fifth text segment is of opponent type ("type="opp""), the input pair with label 1 would be ⟨*"Admittedly the number of immigrants is constantly rising in Germany" ; "but without sufficient, well-qualified junior employees there is hardly a possibility for adequate pension financing."*⟩. The second document of this input pair shows that, although the number of immigrants is increasing, we should not take it for granted since number is not the only factor, the employees qualification and skills is equally important. Even though these two statement are of different types (one supports the main claim and other is against it), the contradiction is barely understandable. It can be explained through the fact that, besides the lack of context, while a rebutter can be seen as an argument for the negation of the proposition under attack, an undercutter does not challenge the validity of a proposition, but challenges the acceptability of an inference between two propositions.

After gathering the input pairs (a total of 1133 examples, 403 positive and 730 negative), we shuffle them and split them in two sets, 80% for training ('train.tsv') and 20% for validation ('dev.tsv').

4.1.5 ArgumentEssays

Argument Annotated Essays[11](Stab and Gurevych, 2017) is a corpus of 402 persuasive essays annotated with discourse-level argumentation structures modeled as a connected tree. The major claim (author's standpoint) is the root node and is usually found in the essay's introduction, that describes the controversial topic. The arguments, presented in individual paragraphs of an essay, can support or attack the major claim. One argument consists of a claim (central component) and, at least, one premise (reason of the argument). Each claim has a stance that can be either "for"

[11] https://www.informatik.tu-darmstadt.de/ukp/research_6/data/argumentation_mining_1/argument_annotated_essays_version_2/index.en.jsp

(supporting argument) or “against" (attacking argument) the major claim. A premise can be used to justify a claim (relation of support) or to refute it (relation of attack).

In contrast to the previous corpus annotation scheme (described in Section 4.1.4) that splits a microtext/answer in different text segments, but does not process them, this corpus annotation has stricter argument component boundary rules, such as ignoring “shell language", phrases like “Another reason is that" or “I am strongly convinced", that are not relevant for the argument’s content.

The corpus has, for each essay, a “.txt" file, that is the entire and unchanged essay, and an annotation file (e.g., “essay001.ann") which contains:

- **Entities -** They can be “MajorClaim", “Claim" or “Premise". One line representing an entity has the entity id, the entity tag, character positions in the essay “.txt" file where the entity starts and ends, and the entity content, as shown in the following example: *T1 MajorClaim 503 575 we should attach more importance to cooperation during primary education*

- **Relations -** They can assume the values “supports" or “attacks". One line representing a relation has the relation id, the relation value, the relation’s source argument/entity id (that can only be of a premise or of a claim), and the relation’s target argument/entity id (that can be of a premise, claim or major claim), as shown in the following example: *R1 supports Arg1:T4 Arg2:T3*

- **Attributes -** They are the claim’s stance and can be “For" or “Against". One line representing an attribute has the attribute id, the tag “Stance", the claim id, and the value, as shown in the following example: *A2 Stance T7 Against*

When building our dataset, we consider the positive examples to be the pairs of major claim and claim with stance “Against", and the pairs of two entities that share a relation of attack. On the other hand, the negative examples will be the pairs of major claim and claim of stance “For", and the pairs of two entities that share a relation of support. After gathering the input pairs (a total of 6673 examples, 715 positive and 5958 negative), we shuffle them and split them in two sets, 80% for training (‘train.tsv’) and 20% for validation (‘dev.tsv’).

4.1.6 W2E

As mentioned in the previous chapter, to confirm that a successful transfer learning is caused by the relevance of document relationships depicted in a source dataset, and not only by the increase of training examples, we collected data from W2E[12] (Hoang et al., 2018). With this dataset we expect to achieve negative transfer learning as the documents relationships are not related with the ones in our baseline datasets. While the baselines present contradiction in a political domain, W2E dataset was designed for topic detection and tracking.

W2E is a Worldwide-Event benchmark dataset for topic detection and tracking, containing 207,722 news articles written in English, from 52 mass media channels (e.g., CNN, BBC, Fox

[12] https://sites.google.com/site/w2edataset/

News, etc.). The news articles cover a large set of 4,501 popular events, within the entire year of 2016, each belonging to one out of 10 categories: “Sport", “Science and technology", “Politics and elections", “Law and crime", “International relations", “Health and medicine", “Disasters and accidents", “Business and economy", “Arts and culture", “Armed conflicts and attacks". To select the events, the authors resorted to Wikipedia’s Current Event portal[13] (WCEP) which contains short summaries of events. They chose the year of 2016 because of the variety of popular long-run stories in that period, such as the US presidential election, UK’s European Union membership referendum, Middle East wars, the Summer Olympics, and disasters in North America.

Since different news sources were used, W2E includes distinct views of the same event. Different news articles regarding the same issue belong to the same topic which is assigned to a more generic category. There is a total of 2,015 topics. W2E provides, for each topic, the topic’s information (id, category, and description) and its correspondent events (date of the event, event’s summary, and search query). The fields in each line are tab-separated. Next, we have an example of a topic and its events:

“

*TOPIC-7 Disasters and accidents *** Kollam temple fire*
2016-04-10 A fire occurs at a Hindu temple in the Kollam district ...
Kollam Kerala India Hindu temple fire
2016-04-11 Five workers from the company that supplied fireworks to the Puttingal Temple ...
worker fireworks Puttingal Temple dead

"

Thus, TOPIC-7 belongs to the “Disasters and accidents" category, its description is “Kollam temple fire", and it has two events (one happening on 2016-04-10, and the other on 2016-04-11).

When building our dataset, we consider the positive examples to be a pair of summaries of two news articles from topics assigned to distinct categories, and from topics assigned to the same category for negative examples. We only extracted data from four categories that seemed more distant regarding the content context (“Sport", “Science and technology", “Politics and elections", “Health and medicine"). For both classes, we have the same distribution for each category, as depicted in Table 4.2.

After gathering the input pairs (a total of 4800 examples, 2400 positive and 2400 negative), we shuffle them and split them in two sets, 80% for training (‘train.tsv’) and 20% for validation (‘dev.tsv’).

4.2 Experimental setup

For each dataset, we train the downstream task by fine-tuning a pre-trained BERT-Base Uncased model (12-layer, 768-hidden, 12-heads, 110M parameters) for sentence-pair classification. When running predictions, the classification model outputs, through a softmax function, a probability distribution over the two possible classes. These probabilities/scores can be seen as the degree of

[13] https://en.wikipedia.org/wiki/Portal:Current_events

Table 4.2: Distribution of text categories in an input pair for each class. The short form for each category is “S" for “Sport", “ST" for “Science and technology", “PE" for “Politics and elections", and “HM" for “Health and medicine".

Class	Categories in a pair	Number of examples
Positive	S-ST	400
	S-PE	400
	S-HM	400
	ST-PE	400
	ST-HM	400
	PE-HM	400
Negative	S-S	600
	ST-ST	600
	PE-PE	600
	HM-HM	600

certainty that a pair of two documents are contradictory or not. Moreover, we consider the highest scored class as being the prediction result.

4.2.1 Environment and tools

The experiments were conducted on the Google Colaboratory platform[14] in order to take advantage of the 12-hours Python 3 runtime with free Colab Cloud Tensor Processing Unit (TPU) hardware acceleration running on the Ubuntu operating system.

We used the BERT modules from the original tensorflow source code[15]. Important things to retain from the repository code are: the model (class BertModel) is implemented in modeling.py; we use the run_classifier.py to construct the classification layer used for the fine-tuning process of the supervised model; and tokenization.py is the tokenizer used to convert the input words into WordPieces, appropriate for BERT. When using the run_classifier to create a classification model, it includes the pooler layer of the pre-trained model to take the representation/embedding of the initial token ([CLS]), that, through multiple attention layers, depends on the rest of the input tokens. It also resorts to the softmax function of TensorFlow’s tf.nn module to perform the classification.

Our models and evaluation and test results were stored in a Google Cloud Storage (GCS) bucket. The datasets used were in a personal Google Drive which we could access by mounting it to Google Colab.

Regarding result analysis, we calculated Precision-Recall, ROC curves and AUC, resorting to ‘precision_recall_curve’, ‘roc_curve’ and ‘auc’ from Scikit-learn’s sklearn.metrics module.

[14] https://colab.research.google.com/notebooks/intro.ipynb
[15] https://github.com/google-research/bert

Table 4.3: Software and respective versions.

Software	Version
Matplotlib	3.2.1
Pandas	1.0.4
Python	3.6.9
Scikit-learn	0.22.2.post1
TensorFlow	1.15.2
Ubuntu	18.04

For plotting, we used the ‘pyplot’ interface from the Python’s library Matplotlib. We also used Pandas ‘crosstab’ to display confusion matrices, and the ‘classification_report’ of Scikit-learn’s sklearn.metrics module to build text reports containing the main classification metrics.

For experimental reproducibility purposes, Table 4.3 presents the versions of the used software.

4.2.2 Train, evaluation and test

In this section we describe the steps we took in order to fine-tune a sentence-pair classification task built on top of a pre-trained BERT model and run predictions on the tuned model.

Table 4.4 shows the used neural network hyperparameters.

Since we are using the Google Cloud TPUs, the optimizer used for fine tuning is tf.contrib.tpu.CrossShardOptimizer, that averages gradients across TPU shards. It encapsulates a basic Adam optimizer that includes “correct" L2 weight decay. This optimization algorithm is a type of Stochastic Gradient Descent[16] with momentum (moving average of the gradient).

One epoch means passing, forward and backward, the entire dataset through the neural network, only in one time. Nevertheless, in practice, we do not feed the entire dataset at once. We rather divide it in several batches and pass them multiple times (more than one epoch). Moreover, using only one epoch would not be enough to update the neural network weights since we are using a type of Gradient Descent which is an iterative learning optimization process.

[16] An iterative optimization algorithm which has a learning rate.

Table 4.4: Model hyperparameters

Hyperparameter	Value
Train batch size	32
Validation batch size	8
Predictive batch size	8
Learning rate	2x10^{-5}
Maximum sequence length	512
Warmup propotion	0.1
Number of training epochs	5

A batch is a set of examples, hence, the train batch size is the total number of training examples in a single batch.

Iteration is the number of batches needed to complete one epoch. Therefore, if we have a dataset of E examples which we divide into batches of E_B examples each, then we need E/E_B iterations to complete one epoch.

A model's checkpoint captures the exact value of all model parameters (tf.Variable objects) and weights. Following the previous definition of iteration, in order to save a tensorflow checkpoint of the model at the end of each epoch, after selecting the dataset, we configure the model with a 'save_checkpoints_steps' (number of steps to complete before saving a checkpoint) equal to the quotient of dividing the number of training examples by the train batch size.

For each dataset we perform the following procedure:

1. Get the train set (a subset of the chosen dataset);
2. Set the model's number of steps required to save a checkpoint;
3. Train the classification model for 5 epochs;
4. Validate the last checkpoint of the model on the development dataset (a subset of the chosen dataset);
5. Choose a dataset to test;
6. Test the model, in its last state/checkpoint.

BERT is a language representation model that was pre-trained on a Wikipedia large corpus in order to learn language patterns. Due to the corpus dimension and diversity, the learned word representations are expected to be generic enough and free from bias that could affect our model performance.

We fine-tune a classification task built on top of a pre-trained BERT model that we load from a saved checkpoint provided by Google in a public Google Cloud storage bucket (gs://cloud-tpu-checkpoints/bert/uncased_L-12_H-768_A-12). This checkpoint was converted from google-research/bert.

To convert data for sequence classification datasets, BERT has four processor classes: Xnli (Cross-Lingual NLI), Mnli (Multi-Genre Natural Language Inference), Mrpc (Microsoft Research Paraphrase Corpus), and Cola (The Corpus of Linguistic Acceptability). These classes extract data into the following parameters: guid (example's unique id), text_a (untokenized text corresponding to the first sequence, and the only required sequence when considering single sentence classification tasks), text_b (optional untokenized text corresponding to the second sequence, and should only be specified in sentence-pair classification tasks), and label (example's label).

Hence, we choose the BERT fine-tuning runner's processor for the Microsoft Research Paraphrase Corpus (MRPC) data set (GLUE version) because this is a corpus of sentence pairs annotated with only two possible labels, 0 or 1. However, we do not use that corpus from the GLUE benchmark. We rather adapt our datasets to the MRPC format so the processor can handle our data. Therefore, as referred in Section 4.1, for a dataset we need to create a Tab Separated Values file for training (“train.tsv"), another for evaluation (“dev.tsv"), and another for testing (“test.tsv"). According to the previous paragraph, we mimic the MRPC data set schema by composing each of these files with five columns: “Quality" (the example label, 0 or 1), “#1 ID" (id of the first document of the example pair), “#2 ID" (id of the second document of the example pair), “#1 String" (the first document of the example pair), and “#2 String" (the second document of the example pair).

4.2.3 Experiments

The conducted experiments are summarized in Table 4.5.

For baseline DonaldTrump, we do a 10-fold cross validation due to its small dimension, while, for the baseline MultiNLIGovernment we opt for a simple train-validation-test split (70% train, 10% validation, 20% test). For each baseline, we also train a binary classifier on the entire set of examples and test on the other baseline dataset.

For the other datasets (MultiNLI-, US2016, ArgumentativeMicrotext, ArgumentEssays, and W2E), we use 80% for training and 20% for validation/evaluation. Then, we run predictions on baseline datasets. We later, for each of these datasets, perform two separate retrainings. So, for one dataset we consider two distinct binary classifiers, both previously trained with that same data, but that will be retrained with different data, from DonaldTrump or MultiNLIGovernment datasets.

Every time we test with the DonaldTrump dataset (except in cases of 10-fold cross validation), we use the entire set. While with MultiNLIGovernment, we use the same test set used for the experiment where we train and test only with examples from the MultiNLIGovernment dataset. That is, we use 20% of the MultiNLIGovernment baseline.

Tables 4.6, 4.7, 4.8, 4.9 and 4.10 present the size of the training, validation and test sets used in each experiment.

In the first experiment (Table 4.6) we focus on the baseline datasets (Section 4.1.1), DonaldTrump and MultiNLIGovernment. We divide this experiment in two parts, 1.a and 1.b.

In the first part (experiment 1.a), we train and test one uncased BERT base model (BERT$_{Base}$) with the DonaldTrump dataset (**experiment 1.a DonaldTrump**), through 10-fold cross validation (10-FCV), and another BERT$_{Base}$ with the MultiNLIGovernment dataset (**experiment 1.a MultiNLIGovernment**). Because for experiment 1.a, in the DonaldTrump dataset we have a limited number of examples and perform 10-FCV, we do not validate the model, so we do not have a development set in this case.

On the other hand, in the second part (experiment 1.b), we train a BERT$_{Base}$ with one baseline dataset, and test the model with a different baseline dataset. For experiment 1.b DonaldTrump, we train a new BERT$_{Base}$ because, this time, since we use the entire DonaldTrump dataset for

Table 4.5: Summary of the conducted experiments and the split ratio of the datasets used in them. The short forms for the (re)training procedures are “10FCV" for “10-fold cross validation", “T-V" for “train-validation split", and “T-V-T" for “train-validation-test split".

Training set	Training procedure	Retraining set	Retraining procedure	Testing set
DonaldTrump	10FCV	-		DonaldTrump
	90%-10% T-V	-		MultiNLIGovernment
MultiNLIGovernment	70%-10% T-V	-		DonaldTrump
	70%-10%-20% T-V-T	-		MultiNLIGovernment
MultiNLI-	80%-20% T-V	-		DonaldTrump
		-		MultiNLIGovernment
		DonaldTrump	10FCV	DonaldTrump
		MultiNLIGovernment	70%-10%-20% T-V-T	MultiNLIGovernment
US2016	80%-20% T-V	-		DonaldTrump
		-		MultiNLIGovernment
		DonaldTrump	10FCV	DonaldTrump
		MultiNLIGovernment	70%-10%-20% T-V-T	MultiNLIGovernment
Argumentative Microtext	80%-20% T-V	-		DonaldTrump
		-		MultiNLIGovernment
		DonaldTrump	10FCV	DonaldTrump
		MultiNLIGovernment	70%-10%-20% T-V-T	MultiNLIGovernment
ArgumentEssays	80%-20% T-V	-		DonaldTrump
		-		MultiNLIGovernment
		DonaldTrump	10FCV	DonaldTrump
		MultiNLIGovernment	70%-10%-20% T-V-T	MultiNLIGovernment
W2E	80%-20% T-V	-		DonaldTrump
		-		MultiNLIGovernment
		DonaldTrump	10FCV	DonaldTrump
		MultiNLIGovernment	70%-10%-20% T-V-T	MultiNLIGovernment

Table 4.6: Dataset size for each baseline experiment. Experiment 1.a is when we train and test a model with the same baseline dataset. Experiment 1.b is when we train in a baseline dataset and test on the other baseline dataset.

	Exp. 1.a DonaldTrump	Exp. 1.a MultiNLIGovernment	Exp. 1.b DonaldTrump	Exp. 1.b MultiNLIGovernment
Train	225	55545	225	-
Dev	-	7935	25	-
Test	25	15870	15870	250

training, we do not resort a 10-FCV, but rather train with a 90%-10% train-validation split approach. Whereas for experiment 1.b MultiNLIGovernment, we load the already trained and validated model from experiment 1.a MultiNLIGovernment, and run prediction on the entire DonaldTrump dataset.

Table 4.7: Dataset size for each experiment 2.a. Experiment 2.a is when we train and validate on a dataset we are exploring, and test on the DonaldTrump baseline dataset.

	Exp. 2.a MultiNLI-	Exp. 2.a US2016	Exp. 2.a ArgumentativeMicrotext	Exp. 2.a ArgumentEssays	Exp. 2.a W2E
Train	266682	1507	907	5339	3840
Dev	66670	375	226	1334	960
Test	250	250	250	250	250

Table 4.8: Dataset size for each experiment 2.b. Experiment 2.b is when we load the trained and validated model from experiment 2.a and test on the MultiNLIGovernment baseline dataset.

	Exp. 2.b MultiNLI-	Exp. 2.b US2016	Exp. 2.b ArgumentativeMicrotext	Exp. 2.b ArgumentEssays	Exp. 2.b W2E
Train	-	-	-	-	-
Dev	-	-	-	-	-
Test	15870	15870	15870	15870	15870

In the second experiment we focus on training a BERT$_{Base}$ model for each dataset we are exploring (Sections 4.1.2, 4.1.3, 4.1.4 and 4.1.5). This experiment is divided in two parts, 2.a and 2.b. In the first part (experiment 2.a, Table 4.7), we train and validate a different BERT$_{Base}$ with each source dataset, MultiNLI- (**experiment 2.a MultiNLI-**), US2016 (**experiment 2.a US2016**), ArgumentativeMicrotext (**experiment 2.a ArgumentativeMicrotext**), ArgumentEssays (**experiment 2.a ArgumentEssays**) and W2E (**experiment 2.a W2E**). Each of these trained models is tested with the entire DonaldTrump dataset. In the rest of this section, we will call these five trained models BERT$_{MultiNLI-}$, BERT$_{US2016}$, BERT$_{Microtext}$, BERT$_{Essays}$ and BERT$_{W2E}$. In the second part (**experiment 2.b**, Table 4.8), we used the trained models from experiment 2.a and run predictions, with each of them, on the MultiNLIGovernment baseline dataset.

Table 4.9: Dataset size for each experiment 3.a. Experiment 3.a is when we load the trained and validated model from experiment 2.a, and retrain it and test on the DonaldTrump baseline dataset.

	Exp. 3.a MultiNLI-	Exp. 3.a US2016	Exp. 3.a ArgumentativeMicrotext	Exp. 3.a ArgumentEssays	Exp. 3.a W2E
Train	225	225	225	225	225
Dev	-	-	-	-	-
Test	25	25	25	25	25

Table 4.10: Dataset size for each experiment 3.b. Experiment 3.b is when we load the trained and validated model from experiment 2.a, and retrain it and test on the MultiNLIGovernment baseline dataset.

	Exp. 3.b MultiNLI-	Exp. 3.b US2016	Exp. 3.b ArgumentativeMicrotext	Exp. 3.b ArgumentEssays	Exp. 3.b W2E
Train	55545	55545	55545	55545	55545
Dev	7935	7935	7935	7935	7935
Test	15870	15870	15870	15870	15870

In the third experiment we focus on retraining and testing the previously trained models, BERT$_{MultiNLI-}$, BERT$_{US2016}$, BERT$_{Microtext}$, BERT$_{Essays}$ and BERT$_{W2E}$, on our baseline datasets. In the first part (**experiment 3.a**), we perform a 10-FCV with each trained model from experiment 2.a and using the baseline DonaldTrump dataset. In the second part (**experiment 3.b**), we also use the trained models from experiment 2.a and train, validate and test them on the MultiNLIGovernment baseline dataset.

4.3 Summary

In this chapter we explained the datasets used in the experiments and the experiments itself.

We detailed each corpus characteristics and how we built our datasets from them. Moreover, we present data examples from those corpora used as source task domains, and compare and relate them with our task of contradiction detection. Nonetheless, we introduce three relations that we are going to study: contradictions in a different domain (across various text genres/topics) , disagreement as contradiction, and arguments of attack as contradictions.

Finally, we describe the project setup and the three conducted experiments: test the baseline datasets (source task domains), explore the new datasets alone, and analyse the new datasets when retrained with examples from our baseline datasets.

Chapter 5

Results and Discussion

In this chapter we reveal the experimental results, analyse and discuss them, and draw conclusions.

Due to the fact that we are facing a binary classification problem, to interpret the results, we will resort to the Receiver Operating Characteristic (ROC) curve, Precision-Recall curve, and Area Under the Curve (AUC) for ROC curve. As said before, when running predictions, our model outputs the probability of an instance (pair of two documents) belonging to each class (contradiction and not contradiction). This way is more flexible since we can interpret those probabilities through different thresholds, contrasting various types of errors, like comparing the number of False Positives (FP) with the number of False Negatives (FN). Indeed, balancing and correlating different measures is important when their costs have distinct impacts.

We also consider the case where the predicted class is the one with higher score/probability (assuming a threshold of 0.5) and, from there, we calculate the number of correct and incorrect predictions, the number of examples predicted as positive or as negative, true positives, false negatives, true negatives, false positives, accuracy, recall, and precision. At the end, we examine the difference between accuracy, recall and precision obtained from the baselines (in experiment 1) and the ones obtained in experiments 2 and 3, in order to see if there were improvements in performance when exploiting new datasets.

5.1 Concepts

This section presents the definition and purpose of some basic concepts important for the following result analysis. Therefore, we describe the metrics and tools used that helped us to interpreting the forecasts of our classification models.

Receiver Operating Characteristic Curve (ROC curve) is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is represented by plotting the **True Positive Rate (TPR)** against the **False Positive Rate (FPR)**, at various threshold[1] settings. As we reduce the threshold value, the predicted condition positive increases, so it is expectable to obtain a higher TPR and higher Recall. However, the FPR also increases, as, in general, we are predicting more examples to be in a positive class. Therefore, specificity (true negative rate, the proportion of actual negatives that are correctly identified) decreases.

ROC is a probability curve and the **Area Under the Curve (AUC)** represents the degree of separability (tells how much a model is capable of distinguishing between classes). For instance, if the AUC is 0.7, this means there is a 70% chance the model will be able to distinguish correctly between positive and negative classes. An excellent model has an AUC near to 1 which means it has a good measure of separability. When the AUC is near 0, the model is predicting 0s (actual negatives) as 1s (positives), and vice versa. Finally, when the AUC is close to 0.5, it means that the model has no class separation capacity whatsoever.

The **early retrieval area** of the ROC plot, where the specificity and threshold are high, can be used to **evaluate high-ranked instances**. A model with a good retrieval level has a bigger AUC in the early retrieval area, meaning that the performance (separability capacity) is better.

The **Precision-Recall curve** shows a trade-off between precision and recall for different thresholds. A high area under the curve represents both high recall (low FNR) and high precision (low FPR). In the case of a bad model, the curve is a horizontal line at the level of the ratio of positive examples existing in the test set. This straight line is called *baseline*. So, if we have a balanced dataset, the baseline would be a straight line of value y = 0.5. On the other hand, a perfect model presents a curve that is the combination of two straight lines, linking the points (0, 1), (1, 1) and (1, y), where y is the baseline (ratio of positive instances). Bad classifiers reach a high recall only at low precision.

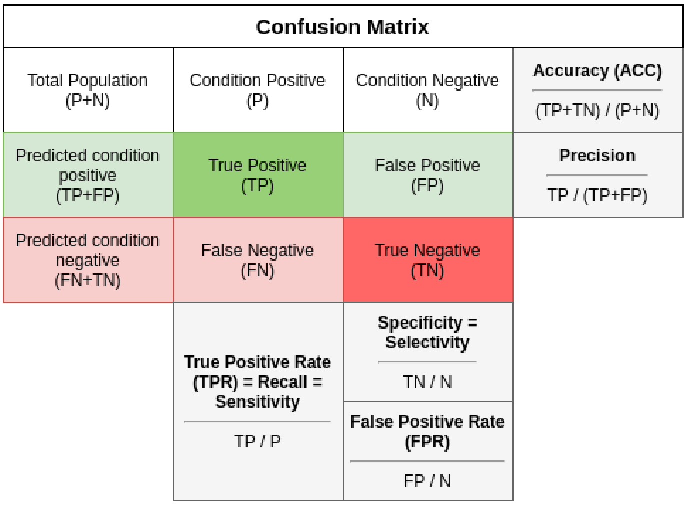
F-1 score is the harmonic mean of precision and recall. When this score has the value of 1 (the higher F-1 score possible) means that the model reaches perfect precision and recall. As we analyse higher threshold values, the precision gets higher because we are looking at the high-scored instances, which are also fewer. In this case we have less positive predictions. Thus, the model has less FP (errors concerning the positive class). However, consequently, the recall is lower due to the same reason, the model predicting fewer positives.

Figure 5.1 shows a scheme of the metrics mentioned above.

5.2 Experiment 1

In this section we describes experiment 1, where we test models performance when using the baseline datasets (the target datasets). For experiment 1.a, we train a model and test it with instances from the same dataset, while for experiment 1.b, the training is done in one baseline and the testing in the other baseline. With this experiment we intent to observe the model capacity for learning contradictions, in the context defined as case study (political domain). Additionally, we

[1] Probability value that must be exceeded by a class score, when classifying an example, in order to predict the example as belonging to that class.



Confusion Matrix

Total Population (P+N)	Condition Positive (P)	Condition Negative (N)	**Accuracy (ACC)** (TP+TN) / (P+N)
Predicted condition positive (TP+FP)	True Positive (TP)	False Positive (FP)	**Precision** TP / (TP+FP)
Predicted condition negative (FN+TN)	False Negative (FN)	True Negative (TN)	
	True Positive Rate (TPR) = Recall = Sensitivity TP / P	**Specificity = Selectivity** TN / N **False Positive Rate (FPR)** FP / N	

Figure 5.1: Confusion Matrix.

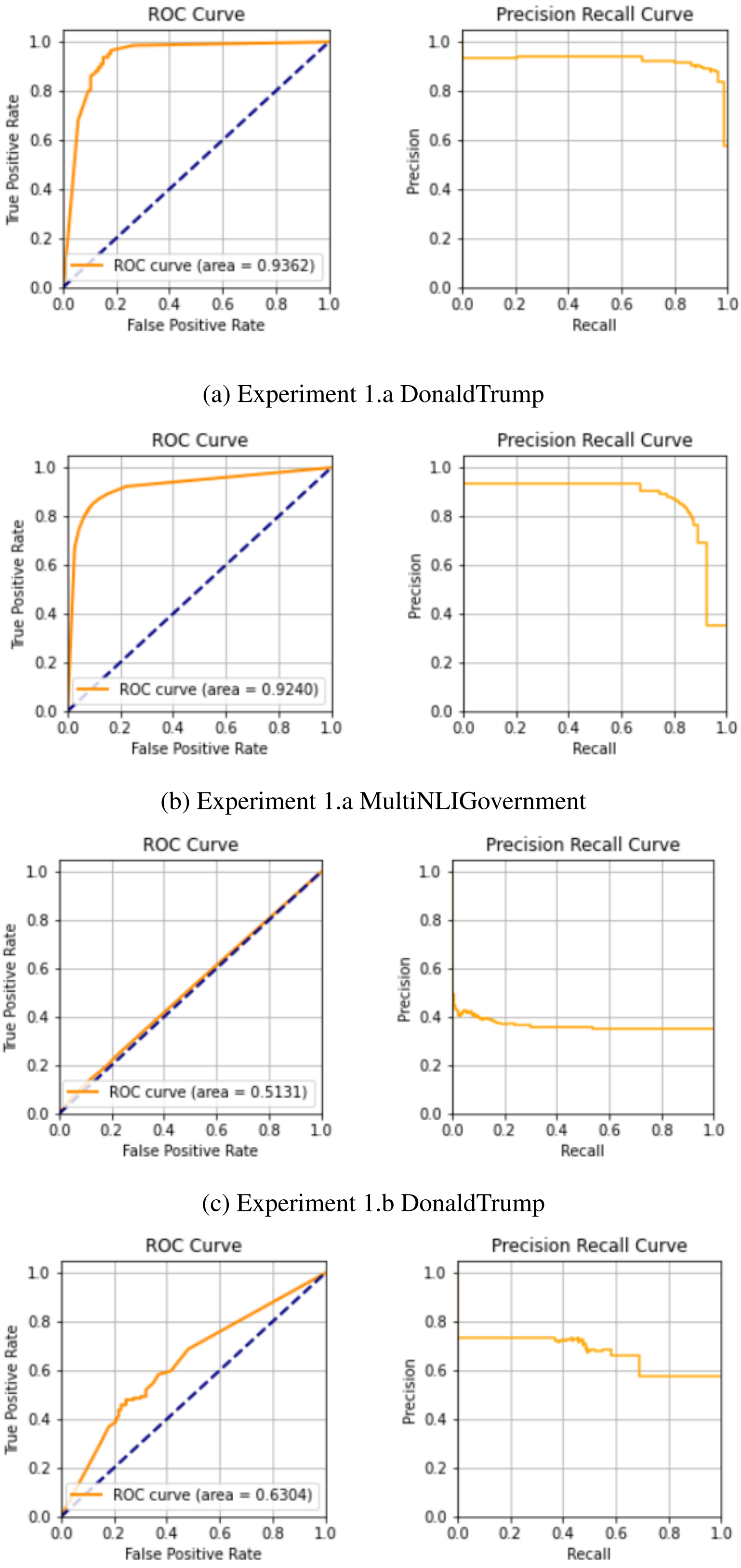
aim to examine how the source of the two baselines impacts the performance when considering simultaneously both datasets for training and testing a model.

As expected, in experiment 1.a MultiNLIGovernment (Figure 5.2b), as we have a large size dataset, the model is able to learn the relation of contradiction between two documents of genre “government". We observe that by the high ROC curve’s AUC and high precision at high recall values. Likewise, for experiment 1.a DonaldTrump (Figure 5.2a), the model is capable of learning the relation of contradiction between two documents of Donald Trump phrases, despite the little number of training examples.

Regarding F1 score, the model in experiment 1.a DonaldTrump can keep a high F1 score around 0.90 through various threshold values (from 0.997 to 0.001), as presented in Appendix B.1.1. For experiment 1.a MultiNLIGovernment, the model can keep its best F1 score of 0.84, with an accuracy of 0.89, through various threshold values (from 0.993 to 0.005), as presented in Appendix B.1.2. These two F1 scores are high which also reflects the separability capability because even when considering very low thresholds, where predicting positive is more frequent, we still have few errors (few false positives). We already achieve the best F1 score at high thresholds in experiment 1.a MultiNLIGovernment, due to the fact of not having a balanced test set (35% of the test examples are positive), while for the DonaldTrump dataset we have 57.6% of positive examples in the test set.

For experiment 1.b, we run predictions on a different dataset, therefore it is expectable to have worse performances than the ones of the previous experiment 1.a.

When training with DonaldTrump dataset and testing with MultiNLIGovernment dataset (Figure 5.2c), we obtain a model that behaves similarly to a random model (that cannot distinguish between the two classes), since the ROC curve coincides with the diagonal. Also, the Precision-Recall curve is close to the baseline, y=0.35, and the best F1 score (Appendix B.2.1) is of only 0.52, at a threshold of 0.003 where all the predictions are positive. We believe that the bad performance is not only due to the dimension difference between train and test sets (very few training



(a) Experiment 1.a DonaldTrump

(b) Experiment 1.a MultiNLIGovernment

(c) Experiment 1.b DonaldTrump

(d) Experiment 1.b MultiNLIGovernment

Figure 5.2: ROC and Precision-Recall curves for each experiment 1 procedure.

Table 5.1: Results for experiment 1 when considering the higher scored label as the predicted class.This table presents the TP, FN, TN, and FP values, and the number of instances predicted as positive examples or as negative examples.

	True Positives	False Negatives	True Negatives	False Positives	Predict Condition Positive	Predict Condition Negative
Experiment 1.a DonaldTrump	13.3	1.1	9	1.6	14.9	10.1
Experiment 1.a MultiNLIGovernment	4597	966	9512	795	5392	10478
Experiment 1.b DonaldTrump	486	5077	9600	707	1193	14677
Experiment 1.b MultiNLIGovernment	68	76	80	26	94	156

examples), but also due to the fact that the test set contains texts from different authors, therefore might contain various language styles, while the train set only has texts from the same person, so is more specific and related to a particular language pattern.

On the other hand, when training with MultiNLIGovernment dataset and testing with DonaldTrump dataset (Figure 5.2d), we have better results than in experiment 1.b DonaldTrump, because we train the model with more examples and larger variety of linguistic markers. Still, the early retrieval area of the ROC curve is small, meaning that the model is not too confident when predicting positive class. Also, the best F1 score (Appendix B.2.2) is of 0.73, but only when classifying all the inputs as positive (at a threshold of 0). Otherwise, through various thresholds, F1 score is around 0.58. This means that both precision and recall are low because we still have a lot of false negatives.

Focusing on Tables 5.1 and 5.2, where we consider the default threshold of 0.5, the recall value for experiment 1.b DonaldTrump is too small, of only 0.087, since the model trained with DonaldTrump examples struggles to classify correctly contradictions. This might be a sign that contradictions in the DonaldTrump scope are well marked by the author's language profile, disabling the model to adapt to a new dataset. The accuracy in experiment 1.b DonaldTrump is slightly better than the one in experiment 1.b MultiNLIGovernment, but that is because the model in the first case is predicting far more negatives than positives (the number of positive predictions is rounded 8.13% of the number of negative predictions), and the test set is not balanced, having more negative examples (the number of positives is rounded 54.0% of the number of negative examples). However, recall and precision are much better in experiment 1.b MultiNLIGovernment.

5.3 Experiment 2

In this section we detail experiment 2, where we train a model on one none-baseline dataset and test it in a different dataset, that is a baseline dataset. Thus, for experiment 2.a we use the baseline

Table 5.2: Results for experiment 1 when considering the higher scored label as the predicted class.This table presents the number of correct and incorrect predictions, accuracy, recall, and precision.

	Correct Predictions	Incorrect Predictions	Accuracy	Recall	Precision
Experiment 1.a DonaldTrump	22.3	2.7	0.892	0.925	0.897
Experiment 1.a MultiNLIGovernment	14109	1761	0.889	0.826	0.853
Experiment 1.b DonaldTrump	10086	5784	0.636	0.087	0.407
Experiment 1.b MultiNLIGovernment	148	102	0.592	0.472	0.723

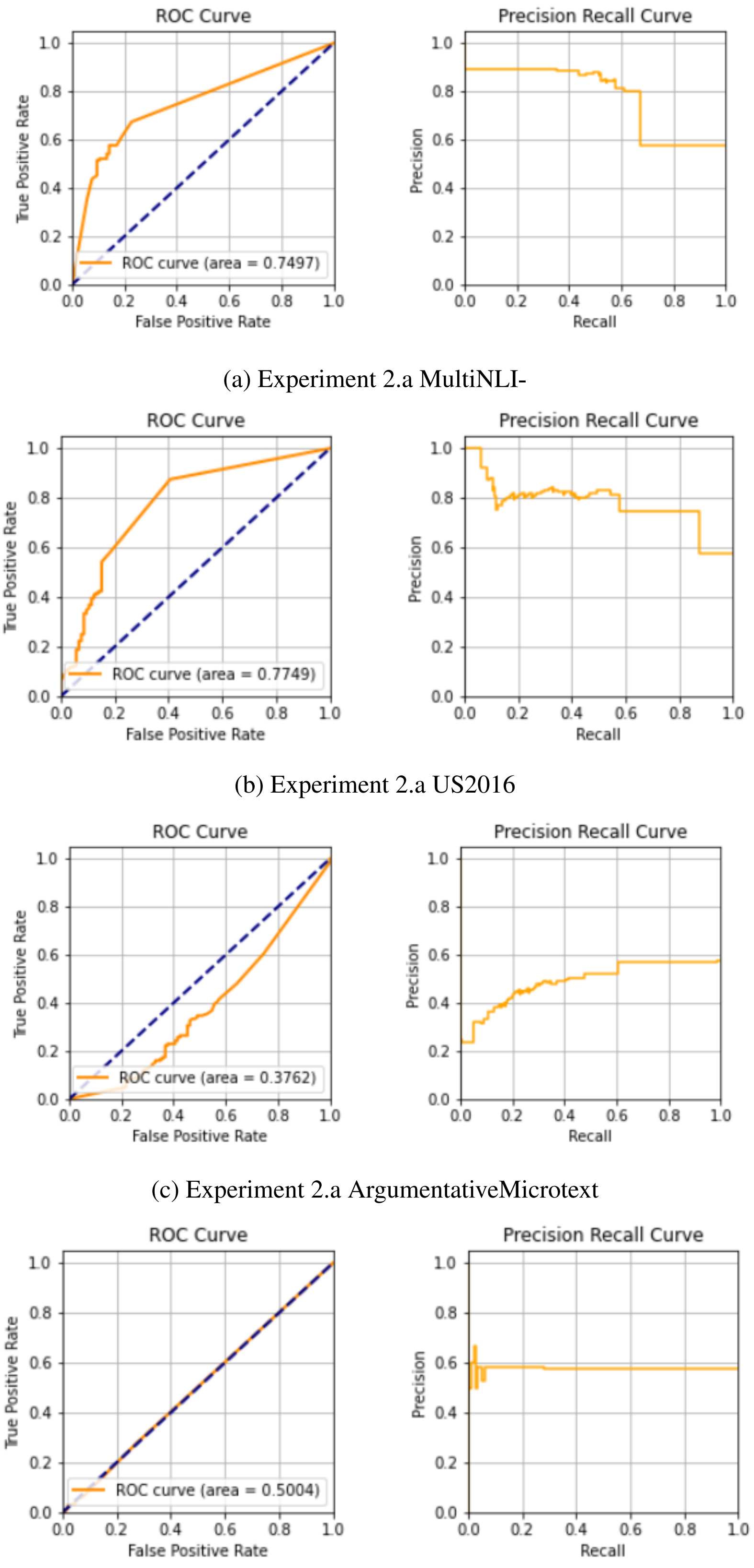
DonaldTrump for running prediction, while in experiment 2.b we use the baseline MultiNLIGovernment. With this experiment we intend to observe if datasets not designed for the same purpose as the baselines (containing relations, between two documents, of contradiction in a political domain) can still be used for the task.

Both experiments 2.a MultiNLI- and 2.a US2016 revealed good results, having a ROC curve's AUC of, respectively, 0.7497 and 0.7749 which means that these two models can distinguish well between the two classes. The early retrieval area of the ROC plot in Figure 5.3a is better than the one in Figure 5.3b, meaning that the model in the first case is more confident when predicting the positive class (true positives with higher score). On the other hand, the model in the second case has better results (bigger ROC curve's AUC) with smaller threshold values. Hence, as we can see in Table 5.4, with a threshold of 0.5, the model in experiment 2.a MultiNLI- performs better than the model in experiment 2.a US2016.

Regarding F1 score, the model in experiment 2.a MultiNLI- reaches its best value of 0.73 at a threshold of 0.001, with an accuracy of 0.72 (Appendix B.3.1). Whereas, the model in experiment 2.a US2016 reaches its best F1 score of 0.81 at the same threshold, with an accuracy of 0.76 (Appendix B.3.2) which confirms that this model is better than the one in experiment 2.a MultiNLI- at really small thresholds. However, experiment 2.a MultiNLI-, through various threshold values, has a F1 score around 0.64, and its lowest F1 score is of 0.51. While, experiment 2.a US2016, through various threshold values, has a F1 score around 0.39, and its lowest F1 score is of 0.12.

A justification for achieving better results in experiment 2.a when resorting to these two datasets might be the fact that the MultiNLI- dataset has examples of real contradiction prepared by professional annotators. Besides that, this dataset has a big dimension. In contrast, the US2016 was not design to contain examples of contradiction, but it is in a political domain and, on top of that, it has examples of Donald Trump speeches which might have helped the model, at some extend, learning language patterns characteristic of this person.

The ArgumentativeMicrotext and ArgumentEssays datasets are not useful for the task since with the latter we obtain a model that behaves similarly to a random model (Figure 5.3d), incapable



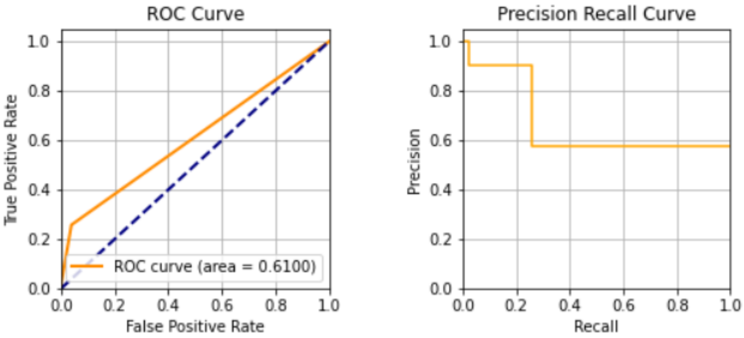
(a) Experiment 2.a MultiNLI-

(b) Experiment 2.a US2016

(c) Experiment 2.a ArgumentativeMicrotext

(d) Experiment 2.a ArgumentEssays

Figure 5.3: ROC and Precision-Recall curves for each experiment 2.a procedure.



(e) Experiment 2.a W2E

Figure 5.3: ROC and Precision-Recall curves for each experiment 2.a procedure (cont.).

Table 5.3: Results for experiment 2.a (test models on baseline DonaldTrump) when considering the higher scored label as the predicted class. This table presents the TP, FN, TN, and FP values, and the number of instances predicted as positive examples or as negative examples.

	True Positives	False Negatives	True Negatives	False Positives	Predict Condition Positive	Predict Condition Negative
Experiment 2.a MultiNLI-	76	68	92	14	90	160
Experiment 2.a US2016	36	108	97	9	45	205
Experiment 2.a ArgumentativeMicrotext	35	109	63	43	78	172
Experiment 2.a ArgumentEssays	1	143	105	1	2	248
Experiment 2.a W2E	0	144	106	0	0	250

Table 5.4: Results for experiment 2.a (test models on baseline DonaldTrump) when considering the higher scored label as the predicted class.This table presents the number of correct and incorrect predictions, accuracy, recall, and precision.

	Correct Predictions	Incorrect Predictions	Accuracy	Recall	Precision
Experiment 2.a MultiNLI-	168	82	0.672	0.528	0.844
Experiment 2.a US2016	133	117	0.532	0.250	0.800
Experiment 2.a ArgumentativeMicrotext	98	152	0.392	0.243	0.449
Experiment 2.a ArgumentEssays	106	144	0.424	0.007	0.500
Experiment 2.a W2E	106	144	0.424	0.000	0.000

Table 5.5: Results for experiment 2.b (test models on baseline MultiNLIGovernment) when considering the higher scored label as the predicted class. This table presents the TP, FN, TN, and FP values, and the number of instances predicted as positive examples or as negative examples.

	True Positives	False Negatives	True Negatives	False Positives	Predict Condition Positive	Predict Condition Negative
Experiment 2.b MultiNLI-	4669	894	9677	630	5299	10571
Experiment 2.b US2016	1236	4327	9706	601	1837	14033
Experiment 2.b ArgumentativeMicrotext	2568	2995	5781	4526	7094	8776
Experiment 2.b ArgumentEssays	54	5509	10215	92	146	15724
Experiment 2.b W2E	25	5538	10281	26	51	15819

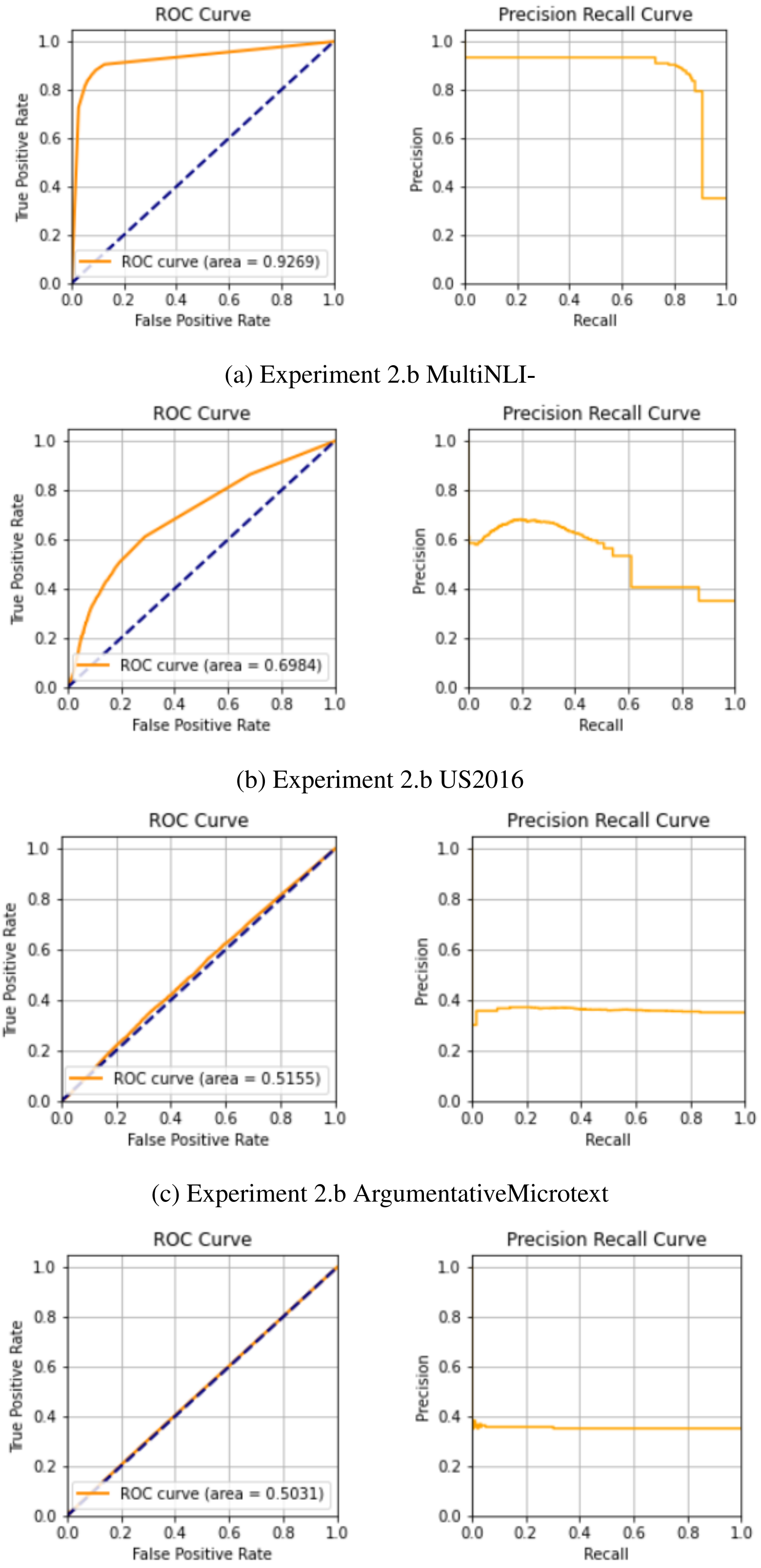
of distinguish between the two classes, while with the first even worst, a model that frequently mistakes positives with negatives, and vice versa (ROC curve's AUC bellow 0.5 in Figure 5.3c).

In Figure 5.3e, the reasonable early retrieval area of the ROC plot might lead us to the wrong conclusion that the model in experiment 2.a W2E has high ranked true positives. However, the threshold considered in that area is already too small, as we can see in Appendix B.3.3 where the first example predicted as positive is when the considered threshold is of 0.3. Indeed, this model is always outputting an high probability for the case of an instance belonging to the negative class (not contradiction). From the beginning, we were expecting a bad performance while using the W2E dataset for training, since it was designed for a task (topic detection) that is not close, whatsoever, to our target task (detecting contradictions). We believe that the model always predicting the negative label is due to the fact that W2E dataset has negative examples that consist of a pair of two documents in a political context (from the category "Politics and election"), therefore, when it receives an example from the testing dataset, DonaldTrump, which is a pair of two texts in a political context, or at least regarding the same topic, the output will be the negative class.

Moreover, the Precision-Recall curves in Figures 5.3c, 5.3d and 5.3e support that the correspondent models are bad since those curves converge towards the baseline, y=0.576, whereas the good models from experiments 2.a MultiNLI- and 2.a US2016 have higher Precision-Recall curve's AUC.

As expected, experiment 2.b MultiNLI- yielded excellent results (in Figure 5.4a both ROC and Precision-Recall curves are close to the ones of a perfect model) because both MultiNLI- and MultiNLIGovernment have examples from the same corpus. Moreover, as the MultiNLI- dataset covers all corpus genres, except the "government" genre, it has bigger dimension which might be the reason why, in this experiment, the model outperforms the baseline, in experiment 1.a MultiNLIGovernment. Though the improvements were not significant.

Experiment 2.b US2016 also showed good results. From Table 5.6 we can see that, considering



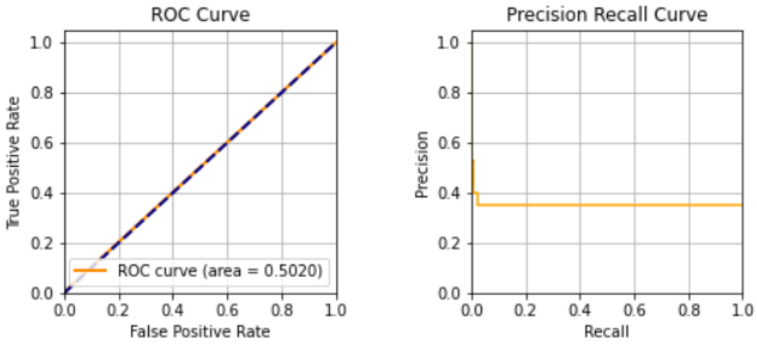
(a) Experiment 2.b MultiNLI-

(b) Experiment 2.b US2016

(c) Experiment 2.b ArgumentativeMicrotext

(d) Experiment 2.b ArgumentEssays

Figure 5.4: ROC and Precision-Recall curves for each experiment 2.b procedure.



(a) Experiment 2.b W2E

Figure 5.5: ROC and Precision-Recall curves for each experiment 2.b procedure (cont.).

Table 5.6: Results for experiment 2.b (test models on baseline MultiNLIGovernment) when considering the higher scored label as the predicted class.This table presents the number of correct and incorrect predictions, accuracy, recall, and precision.

	Correct Predictions	Incorrect Predictions	Accuracy	Recall	Precision
Experiment 2.b MultiNLI-	14346	1524	0.904	0.839	0.881
Experiment 2.b US2016	10942	4928	0.689	0.222	0.673
Experiment 2.b ArgumentativeMicrotext	8349	7521	0.526	0.462	0.362
Experiment 2.b ArgumentEssays	10269	5601	0.647	0.010	0.370
Experiment 2.b W2E	10306	5564	0.649	0.004	0.490

the threshold default value of 0.5, in this experiment we achieved an accuracy and precision of rounded 70%. However, recall has a low value of 22.2%, meaning that the model does not capture a lot of the existing positive cases. The satisfactory outcome here might be due to the fact that US2016 is in a political domain and that a relation of disagreement is, indeed, a good fit for our task of contradiction detection.

Once again, ArgumentativeMicrotext (Figure 5.4c) and ArgumentEssays (Figure 5.4d), present really poor results, similar to the ones of a random model. Furthermore, from Tables 5.3 and 5.5, we can spot that training with ArgumentativeEssays dataset, the model predicts more negatives than positives, leading to low recall values, as a result of being an unbalanced dataset, with more negative instances (the number of positives is rounded 12.0% of the number of negative examples).

The results achieved while using the W2E dataset for training are bad as we anticipated, close to the ones of a random model, and the model is also predicting much more negatives than positives, perhaps due to the same reason mentioned above for experiment 2.a W2E.

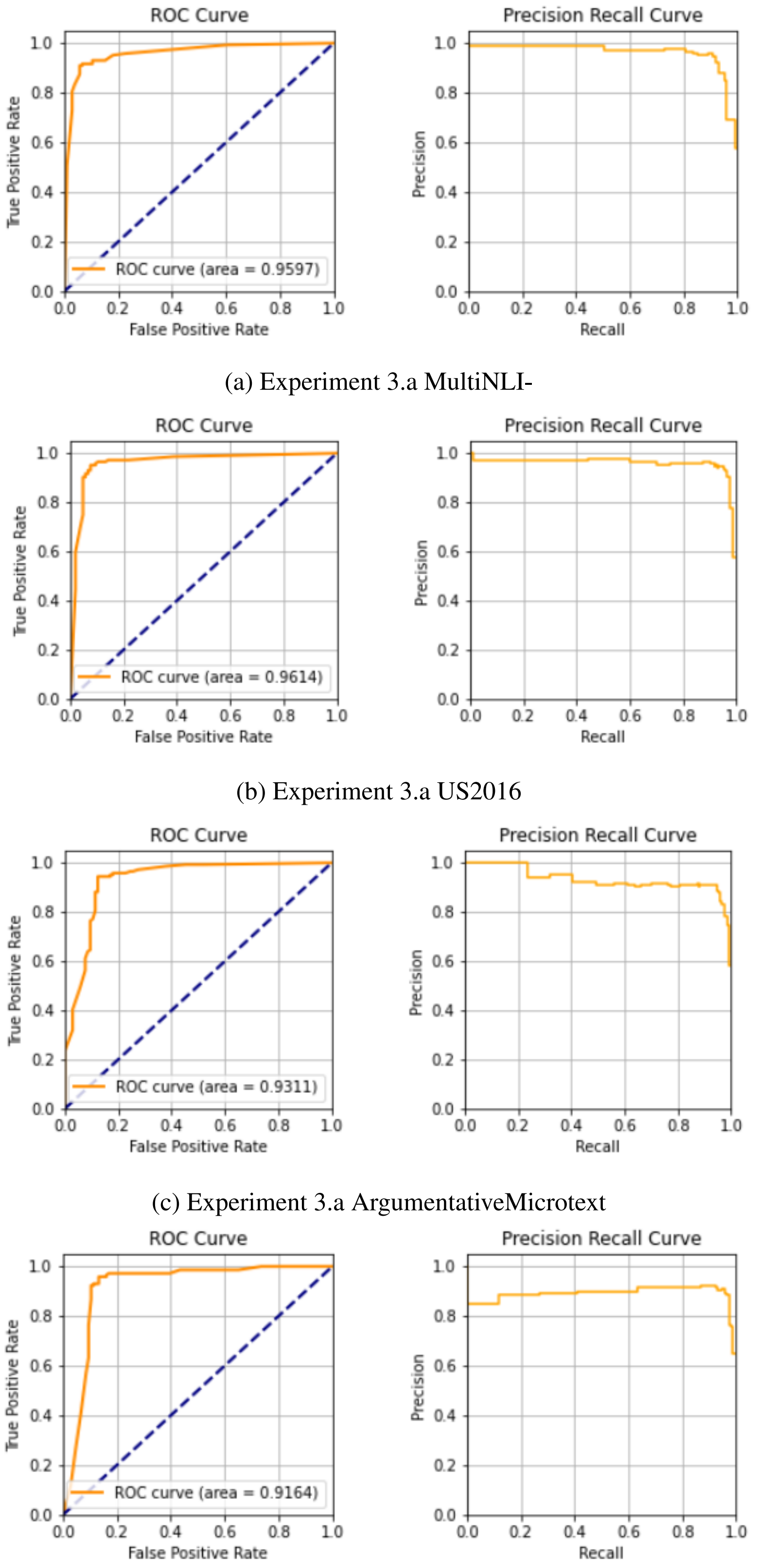
Regarding Precision-Recall curve, the bad models in Figures 5.4c, 5.4d and 5.5a have one that overlaps the the baseline y=0.351. The model in Figure 5.4b, as said before, revealed a better classification performance, and, thus, its Precision-Recall curve has also a higher AUC, but the curve sooner starts to converge towards the baseline. Finally, the Precision-Recall curve from experiment 2.b MultiNLI- reflects the model outstanding classification performance, as the curve is close to the one of a perfect model.

5.4 Experiment 3

In this section we detail experiment 3, where we use previously trained models, from experiment 2, to retrain them and test them again on our baseline datasets. Thus, for experiment 3.a the retraining and testing are done only with instances from the baseline DonaldTrump, while in experiment 3.b those processes are conducted just with examples from MultiNLIGovernment dataset. With this experiment we intend to observe whether we can refine the baseline models with instances of other datasets.

In experiment 3.a, as well as in experiment 3.b, all models, except the ones that also use W2E dataset for training (experiment 3.a W2E and experiment 3.b W2E), revealed high performances, independently of the base model used (trained model from experiment 2). Indeed, there is no significant difference in performance between source datasets used for refining a model that is later retrained and tested with a baseline dataset. Hence, both ROC and Precision-Recall curves in experiments using the datasets MultiNLI- (Figure 5.6a and Figure 5.8a), US2016 (Figure 5.6b and Figure 5.8b), ArgumentativeMicrotext (Figure 5.6c and Figure 5.8c) and ArgumentEssays (Figure 5.6d and Figure 5.8d) are similar to the ones of a perfect model.

On the other hand, when trying to refine a model by using the W2E datatset, not only it did not improve the model or, at least, achieved good prediction results, but also lead to a model which behaviour is similar to the one of a random model. Thus, in this case (Figure 5.7a and Figure 5.9a), both ROC and Precision-Recall curves are similar to the ones of a random classifier



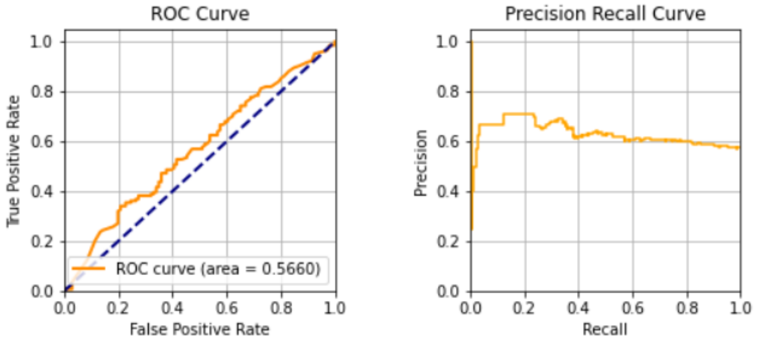
(a) Experiment 3.a MultiNLI-

(b) Experiment 3.a US2016

(c) Experiment 3.a ArgumentativeMicrotext

(d) Experiment 3.a ArgumentEssays

Figure 5.6: ROC and Precision-Recall curves for each experiment 3.a procedure.



(a) Experiment 3.a W2E

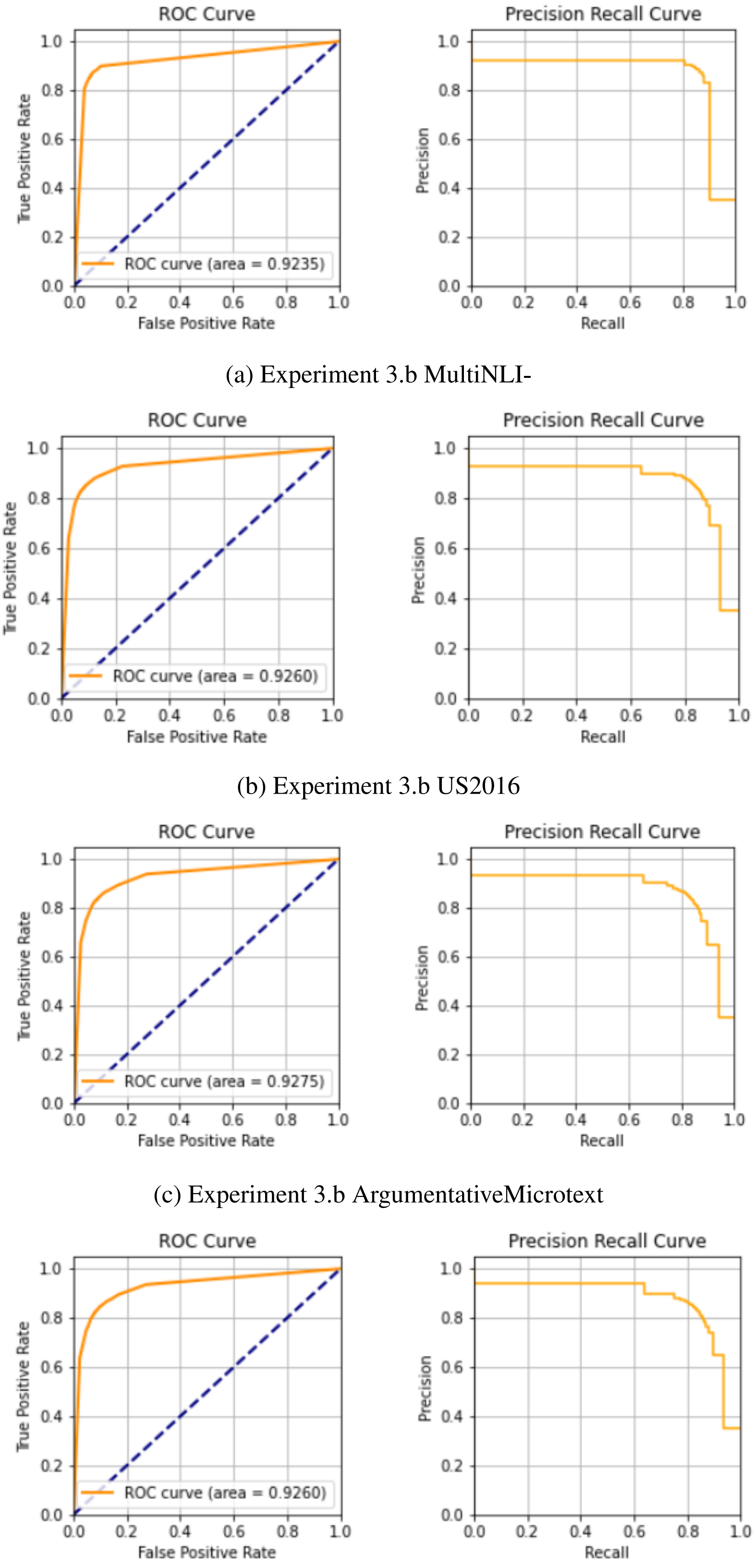
Figure 5.7: ROC and Precision-Recall curves for each experiment 3.a procedure (cont.).

Table 5.7: Results for experiment 3.a (retrain models on baseline DonaldTrump) when considering the higher scored label as the predicted class. This table presents the TP, FN, TN, and FP values, and the number of instances predicted as positive examples or as negative examples.

	True Positives	False Negatives	True Negatives	False Positives	Predict Condition Positive	Predict Condition Negative
Experiment 3.a MultiNLI-	13.2	1.2	9.7	0.9	14.1	10.9
Experiment 3.a US2016	12.9	1.5	10.1	0.5	13.4	11.6
Experiment 3.a ArgumentativeMicrotext	13.6	0.8	9.2	1.4	15	10
Experiment 3.a ArgumentEssays	13.5	0.9	9.2	1.4	14.9	10.1
Experiment 3.a W2E	5.4	9	7.2	3.4	8.8	16.2

Table 5.8: Results for experiment 3.a (retrain models on baseline DonaldTrump) when considering the higher scored label as the predicted class. This table presents the number of correct and incorrect predictions, accuracy, recall, and precision.

	Correct Predictions	Incorrect Predictions	Accuracy	Recall	Precision
Experiment 3.a MultiNLI-	22.9	2.1	0.921	0.921	0.938
Experiment 3.a US2016	23	2	0.920	0.899	0.965
Experiment 3.a ArgumentativeMicrotext	22.8	2.2	0.912	0.948	0.907
Experiment 3.a ArgumentEssays	22.7	2.3	0.908	0.941	0.910
Experiment 3.a W2E	12.6	12.4	0.504	0.377	0.552



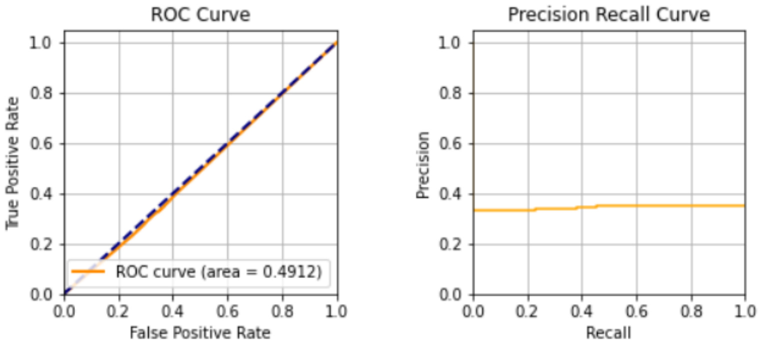
(a) Experiment 3.b MultiNLI-

(b) Experiment 3.b US2016

(c) Experiment 3.b ArgumentativeMicrotext

(d) Experiment 3.b ArgumentEssays

Figure 5.8: ROC and Precision-Recall curves for each experiment 3.b procedure.



(a) Experiment 3.b W2E

Figure 5.9: ROC and Precision-Recall curves for each experiment 3.b procedure (cont.).

Table 5.9: Results for experiment 3.b (retrain models on baseline MultiNLIGovernment) when considering the higher scored label as the predicted class. This table presents the TP, FN, TN, and FP values, and the number of instances predicted as positive examples or as negative examples.

	True Positives	False Negatives	True Negatives	False Positives	Predict Condition Positive	Predict Condition Negative
Experiment 3.b MultiNLI-	4755	808	9696	611	5366	10504
Experiment 3.b US2016	4652	911	9501	806	5458	10412
Experiment 3.b ArgumentativeMicrotext	4606	957	9491	816	5422	10448
Experiment 3.b ArgumentEssays	4594	969	9459	848	5442	10428
Experiment 3.b W2E	1848	3715	6690	3617	5465	10405

Table 5.10: Results for experiment 3.b (retrain models on baseline MultiNLIGovernment) when considering the higher scored label as the predicted class. This table presents the number of correct and incorrect predictions, accuracy, recall, and precision.

	Correct Predictions	Incorrect Predictions	Accuracy	Recall	Precision
Experiment 3.b MultiNLI-	14451	1419	0.911	0.855	0.886
Experiment 3.b US2016	14153	1717	0.892	0.836	0.852
Experiment 3.b ArgumentativeMicrotext	14097	1773	0.888	0.828	0.850
Experiment 3.b ArgumentEssays	14053	1817	0.886	0.826	0.844
Experiment 3.b W2E	8538	7332	0.538	0.332	0.338

Table 5.11: Improvements of using other datasets: the difference (Diff) in accuracy (A), recall (R) and precision (P) between the performance of an experiment 3 model and the performance of an experiment 1.a model.

Experiment 3	Experiment 3 Model Performance			Baseline	Baseline Model Performance			Diff		
	A	R	P		A	R	P	A	R	P
Experiment 3.a MultiNLI-	0.921	0.921	0.938	DonaldTrump	0.892	0.925	0.897	**0.029**	-0.004	**0.041**
Experiment 3.a US2016	0.920	0.899	0.965	DonaldTrump	0.892	0.925	0.897	**0.028**	-0.026	**0.068**
Experiment 3.a Argumentative Microtext	0.912	0.948	0.907	DonaldTrump	0.892	0.925	0.897	**0.020**	**0.023**	**0.010**
Experiment 3.a ArgumentEssays	0.908	0.941	0.910	DonaldTrump	0.892	0.925	0.897	**0.016**	**0.016**	**0.013**
Experiment 3.a W2E	0.504	0.377	0.552	DonaldTrump	0.892	0.925	0.897	-0.388	-0.548	-0.345
Experiment 3.b MultiNLI-	0.911	0.855	0.886	MultiNLI Government	0.889	0.826	0.853	**0.022**	**0.028**	**0.034**
Experiment 3.b US2016	0.892	0.836	0.852	MultiNLI Government	0.889	0.826	0.853	**0.003**	**0.010**	0.000
Experiment 3.b Argumentative Microtext	0.888	0.828	0.850	MultiNLI Government	0.889	0.826	0.853	-0.001	**0.002**	-0.003
Experiment 3.b ArgumentEssays	0.886	0.826	0.844	MultiNLI Government	0.889	0.826	0.853	-0.004	-0.001	-0.008
Experiment 3.b W2E	0.538	0.332	0.338	MultiNLI Government	0.889	0.826	0.853	-0.351	-0.494	-0.514

(ROC and Precision-Recall curves close to, or even overlapping, respectively, the diagonal and the baseline).

Table 5.11 contains the difference (Diff) in accuracy (A), recall (R) and precision (P) between the performance of a baseline model, from experiment 1.a, and the performance of a model, from experiment 3, trained on a source task domain (new dataset) and retrained on a target task domain (baseline dataset). The metrics here were registered based on a threshold of 0.5 (predicted class being the one with higher probability value).

In bold we highlight the improvements which we can notice that are not that relevant. Still, we had more improvements when considering DonaldTrump as the baseline used because this dataset has less examples, so, by adding examples of another dataset, we are increasing the training set which has proved to be helpful. Furthermore, using the ArgumentativeMicrotext and ArgumentEssays datasets, to extend the MultiNLIGovernment dataset, in general, yields worst results, although it is a minimal loss.

In relation to W2E dataset, there were relevant losses in accuracy, recall, and precision, for both target datasets (our baselines), having decreases ranging from 34.5% to 54.8%.

5.5 Discussion

To the best of our knowledge, this project presents a novel approach of transfer learning for the classification task of detecting contradictions. Even though some models' performances have shown no significant differences between the various datasets tested for our purpose, the obtained results are enough to draw interesting conclusions and answer our research questions.

In experiment 1, it is clear that the BERT model is capable of learning very well the relation between two documents, despite the small number of examples in the train set, which happens in our DonaldTrump dataset. When training a model with one baseline dataset and testing with the other, we achieved better results when training with the MultiNLIGovernment dataset. Besides the bigger set dimension, the MultiNLIGovernment contains reports, speeches, letters and press releases from public domain government websites. Therefore, we are training with documents from different authors, feeding the model with inputs featuring a variety of linguistic patterns. Hence, our theory is that the MultiNLIGovernment is more generic, having cases that can be approximated to Donald Trump characteristic speaking behaviours. Whereas, the DonaldTrump dataset is more specific, unable to adapt to the diversity of the other baseline dataset.

In experiment 2, where the train set is in a different feature space than the test set, we obtained better results training with MultiNLI- and US2016 datasets. MultiNLI- does not cover the political domain, but has reliable examples of contradiction. On the other hand, US2016 is in a political domain, and even incorporates transcripts of Donald Trump speeches (singular language patterns) and online reactions from people of varying backgrounds (wide range of language patterns). Nevertheless, the positive class includes examples of disagreement instead of contradiction. The satisfactory outcomes of using these two datasets corroborate our idea that there might be language patterns characteristic of our target task and domain, since these two cases share common features with our baseline datasets: document-pair relation of contradictions, being in the political domain, and texts of a particular person.

From the findings of experiment 2, we were already expecting better knowledge transfer results when using MultiNLI- and US2016 datasets as source datasets. Nonetheless, as in experiment 1.a the two models performances are already outstanding, the improvements of extending the training set with examples from other datasets, in experiment 3, are not significant.

Additionally, experiments 2 and 3 revealed that arguments' relations of support and attack, in both ArgumentativeMicrotext and ArgumentEssays datasets, are not reliable for transfer learning for the task of detecting contradictions, as those two datasets only lead to gains that are too irrelevant, and even to decreases in performance (worst accuracy, precision, and recall).

Finally, we considered the W2E in order to confirm that, if there was any successful transfer learning, such as the gains in accuracy and precision while using MultiNLI- and US2016 datasets in experiment 3, the reason for it would not only be the increased amount of training examples, but rather the documents' relationship present in the used source datasets. Therefore, since W2E contains a document relationship (two texts sharing the same topic/category or not) that is not related to our target task (detecting contradictions), we anticipated a worst learning performance in experiment 3 for this dataset, which ended-up happening. Hence, it confirmed our idea that relationship between documents was the prominent factor in transfer learning. Otherwise, we would have still achieved good classification results when enlarging the training set with examples from W2E.

Chapter 6

Conclusions and Future Work

This dissertation aim was to propose and implement a methodology to detect contradictions, resorting to transfer learning in order to explore whether other datasets, not specifically designed for the task, can be used to improve the model performance.

We presented a method based on transfer learning, supervised learning, and BERT model for text representation for detecting contradictions between two documents. Our aim was to build a system to identify whether two documents are contradictory or not. We addressed the problem of detecting contradictions as a supervised binary classification problem that takes as input a pair of two documents/texts, and outputs a probability for each of the two possible classes. We employed the BERT model for language modelling and fine-tuned it for our sentence-pair task of contradiction detection.

Considering the assumption that the relation between documents implies specific language patterns, we wanted to test the hypothesis that other distinguishing relations between documents may also suggest similar effects on language. Thus, we believe that such can be proved by comparing the performance of a model only trained on a baseline dataset with the performance of another model trained with examples from both a baseline dataset and a new dataset. If the last model yields better classification results, then our hypothesis is verified.

Following the previous paragraph, experiments were conducted on a total of seven datasets, being two of them the baseline datasets, with the purpose of studying if different document relationships, from various datasets, can be exploited for our task of contradiction detection (a successful transfer of knowledge from one domain to another). In the context of transfer learning, the baselines represent the target task domain, and the five remaining datasets the source task domains. Regarding the two baseline datasets, one was built from certain examples of MultiNLI corpus, and the other was manually constructed and annotated, resorting to an online article listing Donald Trump's contradictions. For the five source task domains, we developed datasets from the following publicly available corpora: MultiNLI corpus[1], US2016 corpus[2], Argumentative Microtext

[1] https://www.nyu.edu/projects/bowman/multinli/

[2] http://www.corpora.aifdb.org/US2016

Corpus[3], Argument Annotated Essays corpus[4], and W2E corpus[5].

The experiments were divided in three sets with different objectives: observe the model capability of learning contradictions, considering only the baseline datasets; observe if datasets not designed for the same purpose as the baselines can still be used for the task; and observe whether we can refine the baseline models with instances of other datasets. The obtained results do not show significant improvements when adding the new datasets for transfer learning. Still, they were sufficient for our investigation purpose.

The higher improvements were in precision when the target task domain was based on the DonaldTrump dataset, and the source task domain was based on the MultiNLI- dataset (with an improvement of 4.1%) or the US2016 dataset (with an improvement of 6.8%). We also had increases around 2% and 3% in accuracy and recall with other source task domains, but in general we had more gains while using the DonaldTrump baseline as target. As expected, using as source task domain the W2E dataset, the resultant model yield worst classification results than the models only trained with examples from baseline datasets. In this case, we registered significant losses ranging from 34.5% to 54.8% in accuracy, precision, and recall (considering a threshold value of 0.5).

Based on the findings, we drew conclusions that, when considering transfer learning for the task of detecting contradictions, the documents topic similarity (different document-pair relations, but same domain genre/topic, or same document-pair relations in various domain genres), the relation of disagreement between two propositions, and texts from the same author are relations that should be taking into account when building a source task domain. Indeed, these three factors might help boosting the model learning performance of the target task (detecting contradictions).

With respect to the goals stated in Chapter 1, we consider them as fulfilled, as we gathered the necessary data, presented a methodology to detect contradictions (Chapter 3), and tested it through empirical evaluations (Chapters 4 and 5).

6.1 Answers to the Research Questions

Below we answer to the research questions raised in Chapter 3.

Considering our target task of detecting whether two documents are contradictory or not, ...

$\mathscr{Q}_1$. *... can a classification model be effective when only trained with examples whose document-pair relations are different from the target one (contradictions)?*
In experiment 2, we noticed that when training a model with our MultiNLI- dataset we obtained outstanding results when running predictions with instances of the MultiNLIGovernment dataset. However, the main cause might be the fact that these two datasets where built based on the same corpus. Still, when testing both baseline datasets, DonaldTrump and

[3] http://angcl.ling.uni-potsdam.de/resources/argmicro.html

[4] https://www.informatik.tu-darmstadt.de/ukp/research_6/data/argumentation_mining_1/argument_annotated_essays_version_2/index.en.jsp

[5] https://sites.google.com/site/w2edataset/

MultiNLIGovernment, the positive outcomes achieved considering MuiltiNLI- and US2016 datasets as train set sources, lead us to believe that, yes, different datasets can be used and adapted for our target task. Nonetheless, we still had losses in performance when only considering these two new datasets for training.

$\mathcal{Q}_2$. *... can other examples, that incorporate document-pair relations different from the target one, be used to provide an extra training set of contradictory statements, in order to improve a model learning performance?*
Although the results exposed in Chapter 5 Section 5.4 show that the improvements are not too significant, being the best improvements an increase of 3% in accuracy and 7% in precision (Table 5.11, experiments 3.a MultiNLI- and 3.a US2016), we verify that, indeed, it is possible to improve the model classification performance when retraining it with instances of different datasets, initially designed for other tasks. Therefore, it means we are refining an inference model for contradiction detection through transfer learning.

6.2 Contributions

The main contribution of this entire project are the following:

- The use of neural language representation models to build the language profile of contradictions, in the defined domain (political);

- The use of neural language representation model to build the language profile of a particular entity (Donald Trump, the president of the United States);

- The use of a classification approach, assuming the existence of language patterns characteristic of the relation between two documents under study, to capture whether two documents contradict themselves;

- The use of various datasets to test the adaptability of different relations, between two documents, when applied to our specific task of contradiction detection;

- Building new datasets for the purpose of detecting contradictions, and making them available online[6];

- The application of the proposed method, using examples created and gathered by experts in publicly available corpora, and real-world data, collected and processed by us (Donald Trump contradictions).

[6] https://github.com/BeatrizBaldaia/sentence-pair-contradictions

6.3 Future Work

This research proposes a novel approach to tackle the complex, recurring and relevant problem of detecting contradictions. Therefore, new opportunities, different approaches and possible improvements never cease. In this section we attempt to enumerate some of them.

6.3.1 Improvements

- The proposed method is independent of the neural language representation model used, thus this one can be replaced by another. It is always recommended to choose the state-of-the-art model when implementing the designed approach and we have been witnessing a rapid evolution regarding state-of-the-art in NLP tasks. Therefore, we believe that improvements can be achieved by trying new models.

- One of the baseline datasets used (DonaldTrump dataset) was built manually by us, missing robustness due to the lack of annotation guidelines, and the lack of professional training, preparation and knowledge of how to create a reliable and representative corpus for a specific task. Furthermore, since this dataset was constructed by a single annotator, we do not employ any technique for calculating annotators agreement level, such as the Inter Annotator Agreement (IAA) metric. Besides all the above, the number of examples incorporated in that dataset is small. Thus, polishing, enriching and increasing this dataset is essential.

- Not all datasets used are balanced in terms of positive and negative examples, which can have serious implications in the model learning process and, consequently, in its classification performance. Therefore, this issue should be addressed.

- Since we are testing various relations between documents for transfer learning in a classification problem, not only we can find other datasets illustrating the same proposed relations, as we can also explore new relations not introduced in this report.

- Explore different variations in BERT model hyperparameters in order to improve the learning process.

6.3.2 Extensions

- In our proposed approach, the model is trained to classify a pair of two documents as contradictory (positive class) or not (negative class). We drew our conclusions based on evaluation metrics, such as accuracy, precision, and recall. However, in order to find which exact relation had a major impact, deep interpretations of neural model predictions could be conducted by applying, for example, gradient-based saliency maps and input reduction (to highlight the most important input features).

- This project could be used as a tool to tackle current real life examples and events. For instance, considering a new model developed to contain a certain person profile (extensive database of that person speeches, interviews and posts in social media) that would be used to find similar texts to the ones it would dynamically receive from social media websites (like Twitter[7]) or online collections of that personality's interviews and speeches (like Factbase[8]), our trained models could be used to run predictions with the documents found in the person profile and the ones dynamically received.

[7] https://twitter.com/explore

[8] https://factba.se/

Appendix A

Methodology overview

A scheme of our proposed methodology.

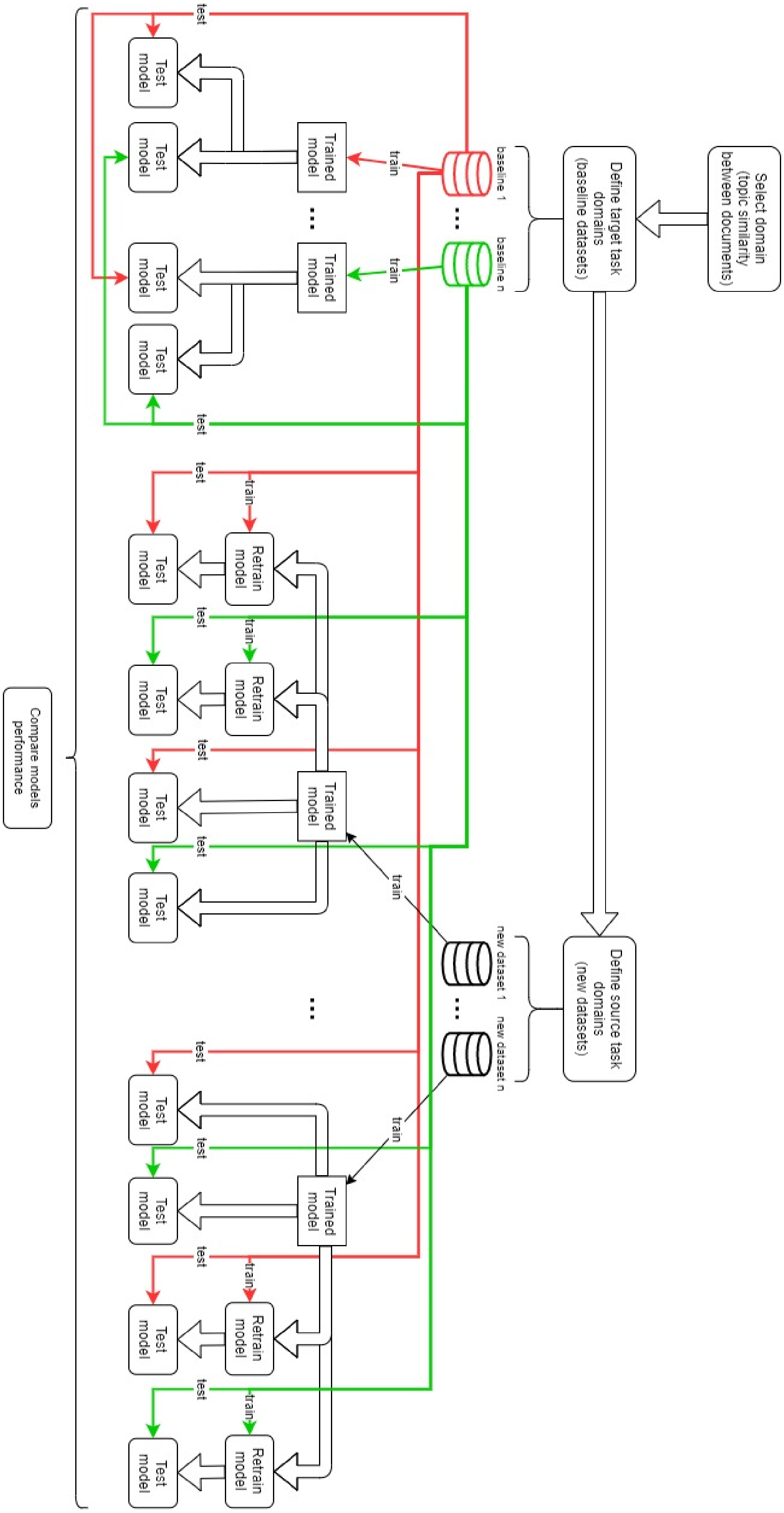


Figure A.1: Methodology overview.

Appendix B

Confusion matrices and classification reports

Confusion matrices and classification reports for various threshold values in conducted experiments.

B.1 Experiment 1.a

B.1.1 DonaldTrump

```
threshold: 1.0
-------------
Predicted    0   1  All
Reality
0          104   2  106
1          114  30  144
All        218  32  250

Classification report:
              precision    recall  f1-score   support

           0       0.48      0.98      0.64       106
           1       0.94      0.21      0.34       144

    accuracy                           0.54       250
   macro avg       0.71      0.59      0.49       250
weighted avg       0.74      0.54      0.47       250
```

```
threshold: 0.999
----------
Predicted     0    1  All
Reality
0           100    6  106
1            46   98  144
All         146  104  250

Classification report:
              precision    recall  f1-score   support

           0       0.68      0.94      0.79       106
           1       0.94      0.68      0.79       144

    accuracy                           0.79       250
   macro avg       0.81      0.81      0.79       250
weighted avg       0.83      0.79      0.79       250


threshold: 0.997
----------
Predicted     0    1  All
Reality
0            95   11  106
1            28  116  144
All         123  127  250

Classification report:
              precision    recall  f1-score   support

           0       0.77      0.90      0.83       106
           1       0.91      0.81      0.86       144

    accuracy                           0.84       250
   macro avg       0.84      0.85      0.84       250
weighted avg       0.85      0.84      0.84       250


threshold: 0.996
```

```
__________
Predicted    0    1  All
Reality
0           95   11  106
1           23  121  144
All        118  132  250

Classification report:
              precision    recall  f1-score   support

           0       0.81      0.90      0.85       106
           1       0.92      0.84      0.88       144

    accuracy                           0.86       250
   macro avg       0.86      0.87      0.86       250
weighted avg       0.87      0.86      0.86       250


threshold: 0.986
__________
Predicted    0    1  All
Reality
0           92   14  106
1           17  127  144
All        109  141  250

Classification report:
              precision    recall  f1-score   support

           0       0.84      0.87      0.86       106
           1       0.90      0.88      0.89       144

    accuracy                           0.88       250
   macro avg       0.87      0.87      0.87       250
weighted avg       0.88      0.88      0.88       250


threshold: 0.979
__________
Predicted    0    1  All
```

```
Reality
0             92   14  106
1             15  129  144
All          107  143  250

Classification report:
              precision    recall  f1-score   support

           0       0.86      0.87      0.86       106
           1       0.90      0.90      0.90       144

    accuracy                           0.88       250
   macro avg       0.88      0.88      0.88       250
weighted avg       0.88      0.88      0.88       250


threshold: 0.206
----------
Predicted   0    1  All
Reality
0          90   16  106
1           9  135  144
All        99  151  250

Classification report:
              precision    recall  f1-score   support

           0       0.91      0.85      0.88       106
           1       0.89      0.94      0.92       144

    accuracy                           0.90       250
   macro avg       0.90      0.89      0.90       250
weighted avg       0.90      0.90      0.90       250


threshold: 0.021
----------
Predicted   0    1  All
Reality
0          88   18  106
```

```
1            9  135  144
All         97  153  250

Classification report:
              precision    recall  f1-score   support

           0       0.91      0.83      0.87       106
           1       0.88      0.94      0.91       144

    accuracy                           0.89       250
   macro avg       0.89      0.88      0.89       250
weighted avg       0.89      0.89      0.89       250


threshold: 0.005
-----------
Predicted    0    1  All
Reality
0           88   18  106
1            7  137  144
All         95  155  250

Classification report:
              precision    recall  f1-score   support

           0       0.93      0.83      0.88       106
           1       0.88      0.95      0.92       144

    accuracy                           0.90       250
   macro avg       0.91      0.89      0.90       250
weighted avg       0.90      0.90      0.90       250


threshold: 0.004
-----------
Predicted    0    1  All
Reality
0           87   19  106
1            7  137  144
All         94  156  250
```

```
Classification report:
              precision    recall  f1-score   support

           0       0.93      0.82      0.87       106
           1       0.88      0.95      0.91       144

    accuracy                           0.90       250
   macro avg       0.90      0.89      0.89       250
weighted avg       0.90      0.90      0.89       250


threshold: 0.002
------------
Predicted   0    1  All
Reality
0          87   19  106
1           5  139  144
All        92  158  250

Classification report:
              precision    recall  f1-score   support

           0       0.95      0.82      0.88       106
           1       0.88      0.97      0.92       144

    accuracy                           0.90       250
   macro avg       0.91      0.89      0.90       250
weighted avg       0.91      0.90      0.90       250


threshold: 0.001
------------
Predicted   0    1  All
Reality
0          78   28  106
1           2  142  144
All        80  170  250

Classification report:
```

```
              precision    recall  f1-score   support

           0       0.97      0.74      0.84       106
           1       0.84      0.99      0.90       144

    accuracy                           0.88       250
   macro avg       0.91      0.86      0.87       250
weighted avg       0.89      0.88      0.88       250


threshold: 0.0
----------
Predicted    1  All
Reality
0          106  106
1          144  144
All        250  250

Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       106
           1       0.58      1.00      0.73       144

    accuracy                           0.58       250
   macro avg       0.29      0.50      0.37       250
weighted avg       0.33      0.58      0.42       250
```

B.1.2 MultiNLIGovernment

```
threshold: 1.0
----------
Predicted      0     1    All
Reality
0          10037   270  10307
1           1829  3734   5563
All        11866  4004  15870

Classification report:
              precision    recall  f1-score   support
```

```
           0       0.85      0.97      0.91     10307
           1       0.93      0.67      0.78      5563

    accuracy                           0.87     15870
   macro avg       0.89      0.82      0.84     15870
weighted avg       0.88      0.87      0.86     15870


threshold: 0.998
----------------
Predicted      0     1    All
Reality
0           9776   531  10307
1           1269  4294   5563
All        11045  4825  15870

Classification report:
              precision    recall  f1-score   support

           0       0.89      0.95      0.92     10307
           1       0.89      0.77      0.83      5563

    accuracy                           0.89     15870
   macro avg       0.89      0.86      0.87     15870
weighted avg       0.89      0.89      0.88     15870


threshold: 0.993
----------------
Predicted      0     1    All
Reality
0           9646   661  10307
1           1097  4466   5563
All        10743  5127  15870

Classification report:
              precision    recall  f1-score   support

           0       0.90      0.94      0.92     10307
           1       0.87      0.80      0.84      5563

    accuracy                           0.89     15870
   macro avg       0.88      0.87      0.88     15870
weighted avg       0.89      0.89      0.89     15870


threshold: 0.509
__________
Predicted      0     1    All
Reality
0           9512   795  10307
1            968  4595   5563
All        10480  5390  15870

Classification report:
              precision    recall  f1-score   support

           0       0.91      0.92      0.92     10307
           1       0.85      0.83      0.84      5563

    accuracy                           0.89     15870
   macro avg       0.88      0.87      0.88     15870
weighted avg       0.89      0.89      0.89     15870


threshold: 0.251
__________
Predicted      0     1    All
Reality
0           9486   821  10307
1            948  4615   5563
All        10434  5436  15870

Classification report:
              precision    recall  f1-score   support

           0       0.91      0.92      0.91     10307
           1       0.85      0.83      0.84      5563
    accuracy                           0.89     15870
   macro avg       0.88      0.87      0.88     15870
weighted avg       0.89      0.89      0.89     15870


threshold: 0.093
----------
Predicted      0     1    All
Reality
0           9466   841  10307
1            925  4638   5563
All        10391  5479  15870

Classification report:
              precision    recall  f1-score   support

           0       0.91      0.92      0.91     10307
           1       0.85      0.83      0.84      5563

    accuracy                           0.89     15870
   macro avg       0.88      0.88      0.88     15870
weighted avg       0.89      0.89      0.89     15870


threshold: 0.005
----------
Predicted     0     1    All
Reality
0          9188  1119  10307
1           764  4799   5563
All        9952  5918  15870

Classification report:
              precision    recall  f1-score   support

           0       0.92      0.89      0.91     10307
           1       0.81      0.86      0.84      5563

    accuracy                           0.88     15870
   macro avg       0.87      0.88      0.87     15870
```

```
weighted avg       0.88      0.88      0.88     15870


threshold: 0.004
------------
Predicted      0     1    All
Reality
0           9124  1183  10307
1            728  4835   5563
All         9852  6018  15870

Classification report:
              precision    recall  f1-score   support

           0       0.93      0.89      0.91     10307
           1       0.80      0.87      0.83      5563

    accuracy                           0.88     15870
   macro avg       0.86      0.88      0.87     15870
weighted avg       0.88      0.88      0.88     15870


threshold: 0.002
------------
Predicted      0     1    All
Reality
0           8771  1536  10307
1            606  4957   5563
All         9377  6493  15870

Classification report:
              precision    recall  f1-score   support

           0       0.94      0.85      0.89     10307
           1       0.76      0.89      0.82      5563

    accuracy                           0.87     15870
   macro avg       0.85      0.87      0.86     15870
weighted avg       0.88      0.87      0.87     15870
```

```
threshold: 0.001
-----------
Predicted      0     1    All
Reality
0           8015  2292  10307
1            429  5134   5563
All         8444  7426  15870

Classification report:
              precision    recall  f1-score   support

           0       0.95      0.78      0.85     10307
           1       0.69      0.92      0.79      5563

    accuracy                           0.83     15870
   macro avg       0.82      0.85      0.82     15870
weighted avg       0.86      0.83      0.83     15870


threshold: 0.0
-----------
Predicted      1    All
Reality
0          10307  10307
1           5563   5563
All        15870  15870

Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00     10307
           1       0.35      1.00      0.52      5563

    accuracy                           0.35     15870
   macro avg       0.18      0.50      0.26     15870
weighted avg       0.12      0.35      0.18     15870
```

B.2 Experiment 1.b

B.2.1 DonaldTrump

```
threshold: 0.997
------------
Predicted      0  1    All
Reality
0          10305  2  10307
1           5560  3   5563
All        15865  5  15870

Classification report:
              precision    recall  f1-score   support

           0       0.65      1.00      0.79     10307
           1       0.60      0.00      0.00      5563

    accuracy                           0.65     15870
   macro avg       0.62      0.50      0.39     15870
weighted avg       0.63      0.65      0.51     15870


threshold: 0.482
------------
Predicted      0     1    All
Reality
0           9594   713  10307
1           5076   487   5563
All        14670  1200  15870

Classification report:
              precision    recall  f1-score   support

           0       0.65      0.93      0.77     10307
           1       0.41      0.09      0.14      5563

    accuracy                           0.64     15870
   macro avg       0.53      0.51      0.46     15870
weighted avg       0.57      0.64      0.55     15870
```

```
threshold: 0.449
------------
Predicted      0     1    All
Reality
0           9583   724  10307
1           5071   492   5563
All        14654  1216  15870

Classification report:
              precision    recall  f1-score   support

           0       0.65      0.93      0.77     10307
           1       0.40      0.09      0.15      5563

    accuracy                           0.63     15870
   macro avg       0.53      0.51      0.46     15870
weighted avg       0.57      0.63      0.55     15870


threshold: 0.28
------------
Predicted      0     1    All
Reality
0           9502   805  10307
1           5028   535   5563
All        14530  1340  15870

Classification report:
              precision    recall  f1-score   support

           0       0.65      0.92      0.77     10307
           1       0.40      0.10      0.16      5563

    accuracy                           0.63     15870
   macro avg       0.53      0.51      0.46     15870
weighted avg       0.56      0.63      0.55     15870
```

```
threshold: 0.154
__________
Predicted      0     1    All
Reality
0           9433   874  10307
1           4984   579   5563
All        14417  1453  15870

Classification report:
              precision    recall  f1-score   support

           0       0.65      0.92      0.76     10307
           1       0.40      0.10      0.17      5563

    accuracy                           0.63     15870
   macro avg       0.53      0.51      0.46     15870
weighted avg       0.56      0.63      0.55     15870



threshold: 0.038
__________
Predicted      0     1    All
Reality
0           9187  1120  10307
1           4841   722   5563
All        14028  1842  15870

Classification report:
              precision    recall  f1-score   support

           0       0.65      0.89      0.76     10307
           1       0.39      0.13      0.20      5563

    accuracy                           0.62     15870
   macro avg       0.52      0.51      0.48     15870
weighted avg       0.56      0.62      0.56     15870


threshold: 0.01
```

```
___________
Predicted      0     1    All
Reality
0           8680  1627  10307
1           4589   974   5563
All        13269  2601  15870

Classification report:
              precision    recall  f1-score   support

           0       0.65      0.84      0.74     10307
           1       0.37      0.18      0.24      5563

    accuracy                           0.61     15870
   macro avg       0.51      0.51      0.49     15870
weighted avg       0.56      0.61      0.56     15870


threshold: 0.007
___________
Predicted      0     1    All
Reality
0           8319  1988  10307
1           4394  1169   5563
All        12713  3157  15870

Classification report:
              precision    recall  f1-score   support

           0       0.65      0.81      0.72     10307
           1       0.37      0.21      0.27      5563

    accuracy                           0.60     15870
   macro avg       0.51      0.51      0.50     15870
weighted avg       0.55      0.60      0.56     15870


threshold: 0.006
___________
Predicted      0     1    All
```

```
Reality
0             8061   2246   10307
1             4233   1330    5563
All          12294   3576   15870

Classification report:
                precision     recall   f1−score    support

            0        0.66       0.78       0.71      10307
            1        0.37       0.24       0.29       5563

     accuracy                              0.59      15870
    macro avg        0.51       0.51       0.50      15870
 weighted avg        0.56       0.59       0.57      15870


threshold: 0.005
-----------
Predicted        0      1      All
Reality
0             7426   2881   10307
1             3898   1665    5563
All          11324   4546   15870

Classification report:
                precision     recall   f1−score    support

            0        0.66       0.72       0.69      10307
            1        0.37       0.30       0.33       5563

     accuracy                              0.57      15870
    macro avg        0.51       0.51       0.51      15870
 weighted avg        0.55       0.57       0.56      15870


threshold: 0.004
-----------
Predicted       0      1      All
Reality
0            4938   5369   10307
```

```
1          2576   2987    5563
All        7514   8356   15870

Classification report:
              precision    recall  f1-score   support

           0       0.66      0.48      0.55     10307
           1       0.36      0.54      0.43      5563

    accuracy                           0.50     15870
   macro avg       0.51      0.51      0.49     15870
weighted avg       0.55      0.50      0.51     15870


threshold: 0.003
__________
Predicted       1     All
Reality
0           10307   10307
1            5563    5563
All         15870   15870

Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00     10307
           1       0.35      1.00      0.52      5563

    accuracy                           0.35     15870
   macro avg       0.18      0.50      0.26     15870
weighted avg       0.12      0.35      0.18     15870
```

B.2.2 MultiNLIGovernment

```
threshold: 1.0
__________
Predicted     0    1   All
Reality
0            90   16   106
1           100   44   144
```

```
All          190   60   250

Classification report:
                   precision     recall   f1-score    support

              0         0.47       0.85       0.61        106
              1         0.73       0.31       0.43        144

       accuracy                               0.54        250
      macro avg         0.60       0.58       0.52        250
   weighted avg         0.62       0.54       0.51        250


threshold: 0.979
-----------
Predicted      0    1   All
Reality
0             83   23   106
1             81   63   144
All          164   86   250

Classification report:
                   precision     recall   f1-score    support

              0         0.51       0.78       0.61        106
              1         0.73       0.44       0.55        144

       accuracy                               0.58        250
      macro avg         0.62       0.61       0.58        250
   weighted avg         0.64       0.58       0.58        250


threshold: 0.723
-----------
Predicted      0    1   All
Reality
0             80   26   106
1             78   66   144
All          158   92   250
```

```
Classification report:
              precision    recall  f1-score   support

           0       0.51      0.75      0.61       106
           1       0.72      0.46      0.56       144

    accuracy                           0.58       250
   macro avg       0.61      0.61      0.58       250
weighted avg       0.63      0.58      0.58       250


threshold: 0.492
----------
Predicted    0   1  All
Reality
0           80  26  106
1           75  69  144
All        155  95  250

Classification report:
              precision    recall  f1-score   support

           0       0.52      0.75      0.61       106
           1       0.73      0.48      0.58       144

    accuracy                           0.60       250
   macro avg       0.62      0.62      0.60       250
weighted avg       0.64      0.60      0.59       250


threshold: 0.395
----------
Predicted    0   1  All
Reality
0           78  28  106
1           75  69  144
All        153  97  250

Classification report:
              precision    recall  f1-score   support
           0       0.51      0.74      0.60       106
           1       0.71      0.48      0.57       144

    accuracy                           0.59       250
   macro avg       0.61      0.61      0.59       250
weighted avg       0.63      0.59      0.59       250


threshold: 0.097
__________
Predicted    0   1  All
Reality
0           77  29  106
1           75  69  144
All        152  98  250

Classification report:
              precision    recall  f1-score   support

           0       0.51      0.73      0.60       106
           1       0.70      0.48      0.57       144

    accuracy                           0.58       250
   macro avg       0.61      0.60      0.58       250
weighted avg       0.62      0.58      0.58       250


threshold: 0.041
__________
Predicted    0   1  All
Reality
0           77  29  106
1           74  70  144
All        151  99  250

Classification report:
              precision    recall  f1-score   support

           0       0.51      0.73      0.60       106
           1       0.71      0.49      0.58       144

    accuracy                           0.59       250
   macro avg       0.61      0.61      0.59       250
weighted avg       0.62      0.59      0.59       250


threshold: 0.01
----------
Predicted    0    1  All
Reality
0           74   32  106
1           73   71  144
All        147  103  250

Classification report:
              precision    recall  f1-score   support

           0       0.50      0.70      0.58       106
           1       0.69      0.49      0.57       144

    accuracy                           0.58       250
   macro avg       0.60      0.60      0.58       250
weighted avg       0.61      0.58      0.58       250


threshold: 0.005
----------
Predicted    0    1  All
Reality
0           72   34  106
1           69   75  144
All        141  109  250

Classification report:
              precision    recall  f1-score   support

           0       0.51      0.68      0.58       106
           1       0.69      0.52      0.59       144
    accuracy                           0.59       250
   macro avg       0.60      0.60      0.59       250
weighted avg       0.61      0.59      0.59       250


threshold: 0.004
-----------
Predicted    0    1  All
Reality
0           69   37  106
1           65   79  144
All        134  116  250

Classification report:
              precision    recall  f1-score   support

           0       0.51      0.65      0.58       106
           1       0.68      0.55      0.61       144

    accuracy                           0.59       250
   macro avg       0.60      0.60      0.59       250
weighted avg       0.61      0.59      0.59       250


threshold: 0.003
-----------
Predicted    0    1  All
Reality
0           67   39  106
1           60   84  144
All        127  123  250

Classification report:
              precision    recall  f1-score   support

           0       0.53      0.63      0.58       106
           1       0.68      0.58      0.63       144

    accuracy                           0.60       250
   macro avg       0.61      0.61      0.60       250
```

```
weighted avg       0.62      0.60      0.61       250


threshold: 0.001
----------
Predicted    0    1  All
Reality
0           55   51  106
1           45   99  144
All        100  150  250

Classification report:
              precision    recall  f1-score   support

           0       0.55      0.52      0.53       106
           1       0.66      0.69      0.67       144

    accuracy                           0.62       250
   macro avg       0.60      0.60      0.60       250
weighted avg       0.61      0.62      0.61       250


threshold: 0.0
----------
Predicted    1  All
Reality
0          106  106
1          144  144
All        250  250

Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       106
           1       0.58      1.00      0.73       144

    accuracy                           0.58       250
   macro avg       0.29      0.50      0.37       250
weighted avg       0.33      0.58      0.42       250
```

B.3 Experiment 2.a

B.3.1 MultiNLI

```
threshold: 1.0
----------
Predicted    0   1  All
Reality
0          100   6  106
1           93  51  144
All        193  57  250

Classification report:
              precision    recall  f1-score   support

           0       0.52      0.94      0.67       106
           1       0.89      0.35      0.51       144

    accuracy                           0.60       250
   macro avg       0.71      0.65      0.59       250
weighted avg       0.74      0.60      0.58       250


threshold: 0.985
----------
Predicted    0   1  All
Reality
0           96  10  106
1           70  74  144
All        166  84  250

Classification report:
              precision    recall  f1-score   support

           0       0.58      0.91      0.71       106
           1       0.88      0.51      0.65       144

    accuracy                           0.68       250
   macro avg       0.73      0.71      0.68       250
weighted avg       0.75      0.68      0.67       250
```

```
threshold: 0.873
----------
Predicted    0   1  All
Reality
0           92  14  106
1           69  75  144
All        161  89  250

Classification report:
              precision    recall  f1-score   support

           0       0.57      0.87      0.69       106
           1       0.84      0.52      0.64       144

    accuracy                           0.67       250
   macro avg       0.71      0.69      0.67       250
weighted avg       0.73      0.67      0.66       250


threshold: 0.056
----------
Predicted    0   1  All
Reality
0           92  14  106
1           66  78  144
All        158  92  250

Classification report:
              precision    recall  f1-score   support

           0       0.58      0.87      0.70       106
           1       0.85      0.54      0.66       144

    accuracy                           0.68       250
   macro avg       0.72      0.70      0.68       250
weighted avg       0.74      0.68      0.68       250
```

```
threshold: 0.005
----------
Predicted    0   1  All
Reality
0           91  15  106
1           61  83  144
All        152  98  250

Classification report:
              precision    recall  f1-score   support

           0       0.60      0.86      0.71       106
           1       0.85      0.58      0.69       144

    accuracy                           0.70       250
   macro avg       0.72      0.72      0.70       250
weighted avg       0.74      0.70      0.69       250


threshold: 0.002
----------
Predicted    0    1  All
Reality
0           86   20  106
1           56   88  144
All        142  108  250

Classification report:
              precision    recall  f1-score   support

           0       0.61      0.81      0.69       106
           1       0.81      0.61      0.70       144

    accuracy                           0.70       250
   macro avg       0.71      0.71      0.70       250
weighted avg       0.73      0.70      0.70       250


threshold: 0.001
----------
```

```
Predicted     0    1  All
Reality
0            82   24  106
1            47   97  144
All         129  121  250

Classification report:
              precision    recall  f1-score   support

           0       0.64      0.77      0.70       106
           1       0.80      0.67      0.73       144

    accuracy                           0.72       250
   macro avg       0.72      0.72      0.71       250
weighted avg       0.73      0.72      0.72       250


threshold: 0.0
—————————
Predicted     1  All
Reality
0           106  106
1           144  144
All         250  250

Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       106
           1       0.58      1.00      0.73       144

    accuracy                           0.58       250
   macro avg       0.29      0.50      0.37       250
weighted avg       0.33      0.58      0.42       250
```

B.3.2 US2016

```
threshold: 0.999
—————————
Predicted     0  1  All
```

```
Reality
0            106   0   106
1            135   9   144
All          241   9   250

Classification report:
               precision    recall  f1-score   support

           0        0.44      1.00      0.61       106
           1        1.00      0.06      0.12       144

    accuracy                            0.46       250
   macro avg        0.72      0.53      0.36       250
weighted avg        0.76      0.46      0.33       250


threshold: 0.981
--------------------
Predicted      0    1   All
Reality
0            100    6   106
1            126   18   144
All          226   24   250

Classification report:
               precision    recall  f1-score   support

           0        0.44      0.94      0.60       106
           1        0.75      0.12      0.21       144

    accuracy                            0.47       250
   macro avg        0.60      0.53      0.41       250
weighted avg        0.62      0.47      0.38       250


threshold: 0.973
--------------------
Predicted      0    1   All
Reality
0            100    6   106
```

```
1           122   22   144
All         222   28   250

Classification report:
              precision    recall  f1-score   support

           0       0.45      0.94      0.61       106
           1       0.79      0.15      0.26       144

    accuracy                           0.49       250
   macro avg       0.62      0.55      0.43       250
weighted avg       0.64      0.49      0.41       250


threshold: 0.94
-----------
Predicted     0    1   All
Reality
0           100    6   106
1           117   27   144
All         217   33   250

Classification report:
              precision    recall  f1-score   support

           0       0.46      0.94      0.62       106
           1       0.82      0.19      0.31       144

    accuracy                           0.51       250
   macro avg       0.64      0.57      0.46       250
weighted avg       0.67      0.51      0.44       250


threshold: 0.745
-----------
Predicted     0    1   All
Reality
0            99    7   106
1           112   32   144
All         211   39   250
```

```
Classification report:
              precision    recall  f1-score   support

           0       0.47      0.93      0.62       106
           1       0.82      0.22      0.35       144

    accuracy                           0.52       250
   macro avg       0.64      0.58      0.49       250
weighted avg       0.67      0.52      0.47       250


threshold: 0.506
----------
Predicted    0   1  All
Reality
0           97   9  106
1          108  36  144
All        205  45  250

Classification report:
              precision    recall  f1-score   support

           0       0.47      0.92      0.62       106
           1       0.80      0.25      0.38       144

    accuracy                           0.53       250
   macro avg       0.64      0.58      0.50       250
weighted avg       0.66      0.53      0.48       250


threshold: 0.124
----------
Predicted    0   1  All
Reality
0           97   9  106
1           96  48  144
All        193  57  250

Classification report:
```

```
              precision    recall  f1-score   support

           0       0.50      0.92      0.65       106
           1       0.84      0.33      0.48       144

    accuracy                           0.58       250
   macro avg       0.67      0.62      0.56       250
weighted avg       0.70      0.58      0.55       250


threshold: 0.01
-----------
Predicted    0   1  All
Reality
0           92  14  106
1           84  60  144
All        176  74  250

Classification report:
              precision    recall  f1-score   support

           0       0.52      0.87      0.65       106
           1       0.81      0.42      0.55       144

    accuracy                           0.61       250
   macro avg       0.67      0.64      0.60       250
weighted avg       0.69      0.61      0.59       250


threshold: 0.004
-----------
Predicted    0   1  All
Reality
0           90  16  106
1           73  71  144
All        163  87  250

Classification report:
              precision    recall  f1-score   support
           0       0.55      0.85      0.67       106
           1       0.82      0.49      0.61       144

    accuracy                           0.64       250
   macro avg       0.68      0.67      0.64       250
weighted avg       0.70      0.64      0.64       250


threshold: 0.003
----------
Predicted    0   1  All
Reality
0           90  16  106
1           66  78  144
All        156  94  250

Classification report:
              precision    recall  f1-score   support

           0       0.58      0.85      0.69       106
           1       0.83      0.54      0.66       144

    accuracy                           0.67       250
   macro avg       0.70      0.70      0.67       250
weighted avg       0.72      0.67      0.67       250


threshold: 0.001
----------
Predicted   0    1  All
Reality
0          63   43  106
1          18  126  144
All        81  169  250

Classification report:
              precision    recall  f1-score   support

           0       0.78      0.59      0.67       106
           1       0.75      0.88      0.81       144
    accuracy                           0.76       250
   macro avg       0.76      0.73      0.74       250
weighted avg       0.76      0.76      0.75       250


threshold: 0.0
————————
Predicted    1  All
Reality
0          106  106
1          144  144
All        250  250

Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       106
           1       0.58      1.00      0.73       144

    accuracy                           0.58       250
   macro avg       0.29      0.50      0.37       250
weighted avg       0.33      0.58      0.42       250
```

B.3.3 W2E

```
threshold: 1.3
————————
Predicted    0  All
Reality
0          106  106
1          144  144
All        250  250

Classification report:
              precision    recall  f1-score   support

           0       0.42      1.00      0.60       106
           1       0.00      0.00      0.00       144
```

```
    accuracy                           0.42       250
   macro avg       0.21      0.50      0.30       250
weighted avg       0.18      0.42      0.25       250


threshold: 0.3
----------
Predicted    0  1  All
Reality
0          106  0  106
1          143  1  144
All        249  1  250

Classification report:
              precision    recall  f1-score   support

           0       0.43      1.00      0.60       106
           1       1.00      0.01      0.01       144

    accuracy                           0.43       250
   macro avg       0.71      0.50      0.31       250
weighted avg       0.76      0.43      0.26       250


threshold: 0.002
----------
Predicted    0  1  All
Reality
0          106  0  106
1          141  3  144
All        247  3  250

Classification report:
              precision    recall  f1-score   support

           0       0.43      1.00      0.60       106
           1       1.00      0.02      0.04       144

    accuracy                           0.44       250
   macro avg       0.71      0.51      0.32       250
```

```
weighted avg        0.76      0.44      0.28       250


threshold: 0.001
-----------
Predicted    0   1  All
Reality
0          102   4  106
1          107  37  144
All        209  41  250

Classification report:
              precision    recall  f1-score   support

           0       0.49      0.96      0.65       106
           1       0.90      0.26      0.40       144

    accuracy                           0.56       250
   macro avg       0.70      0.61      0.52       250
weighted avg       0.73      0.56      0.50       250


threshold: 0.0
-----------
Predicted    1  All
Reality
0          106  106
1          144  144
All        250  250

Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       106
           1       0.58      1.00      0.73       144

    accuracy                           0.58       250
   macro avg       0.29      0.50      0.37       250
weighted avg       0.33      0.58      0.42       250
```

References

Ahmed Al-Rawi. Viral news on social media. *Digital Journalism*, 7(1):63–79, 2019. doi: 10.1080/21670811.2017.1387062. URL https://doi.org/10.1080/21670811.2017.1387062.

Hunt Allcott, Matthew Gentzkow, and Chuan Yu. Trends in the diffusion of misinformation on social media. *Research & Politics*, 6(2):2053168019848554, 2019. doi: 10.1177/2053168019848554. URL https://doi.org/10.1177/2053168019848554.

Sergio A. Alvarez. An exact analytical relation among recall , precision , and classification accuracy in information retrieval. 2002.

I. Androutsopoulos and P. Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187, May 2010. ISSN 1076-9757. doi: 10.1613/jair.2985. URL http://dx.doi.org/10.1613/jair.2985.

N. A. Asad, M. A. Mahmud Pranto, S. Afreen, and M. M. Islam. Depression detection by analyzing social media posts of user. In *2019 IEEE International Conference on Signal Processing, Information, Communication Systems (SPICSCON)*, pages 13–17, 2019.

Ismail Badache, Sébastien Fournier, and Adrian-Gabriel Chifu. Contradiction in reviews: Is it strong or low? In *BroDyn@ECIR*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

Amir Bakarov. A survey of word embeddings evaluation methods, 2018.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. ISSN 01628828. doi: 10.1109/TPAMI.2013.50.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi: 10.1162/tacl_a_00051. URL https://www.aclweb.org/anthology/Q17-1010.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

Katarzyna Budzynska and Chris Reed. Whence inference. *University of Dundee Technical Report*, 2011.

Zhigang Chen, Wei Lin, Qian Chen, Xiaoping Chen, Si Wei, Hui Jiang, and Xiaodan Zhu. Revisiting word embedding for contrasting meaning. 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 106–115, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1011. URL https://www.aclweb.org/anthology/P15-1011.

Timothy Chklovski and Patrick Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *EMNLP*, 2004.

Chenhui Chu and Rui Wang. A survey of domain adaptation for neural machine translation, 2018.

Sarah Cohen, Chengkai Li, Jun Yang, and Cong Yu. Computational journalism: A call to arms to database researchers. pages 148–151, 04 2011.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390177. URL https://doi.org/10.1145/1390156.1390177.

Andrew M. Dai and Quoc V. Le. Semi-supervised sequence learning, 2015.

Hong-Jie DAI, Chi-Yang WU, Richard Tzong-Han TSAI, and Wen-Lian HSU. From entity recognition to entity linking: A survey of advanced entity linking techniques. *Proceedings of the National Congress of the Society for Artificial Intelligence*, JSAI2012:3M2IOS3b1–3M2IOS3b1, 2012. doi: 10.11517/pjsai.JSAI2012.0_3M2IOS3b1.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 193–200, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273521. URL https://doi.org/10.1145/1273496.1273521.

Dipanjan Das, Chris Dyer, Manaal Faruqui, and Yulia Tsvetkov, editors. *Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP*, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-12. URL https://www.aclweb.org/anthology/W16-1200.

Jacob Devlin and Ming-Wei Chang. Open sourcing bert: State-of-the-art pre-training for natural language processing, Nov 2018. URL https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

Valentina Dragos. Detection of contradictions by relation matching and uncertainty assessment. *Procedia Computer Science*, 112:71 – 80, 2017. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2017.08.028. URL http://www.sciencedirect.com/science/article/pii/S1877050917313674. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France.

Ali Farghaly and Khaled Shaalan. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing*, 8(4), December 2009. ISSN 1530-0226. doi: 10.1145/1644879.1644881. URL https://doi.org/10.1145/1644879.1644881.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA, 1998. ISBN 978-0-262-06197-1.

Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171–175, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P12-2034.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-burch. Ppdb: The paraphrase database. In *In HLT-NAACL 2013*, 2013.

Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *KDD 2008 - Proceedings of the 14th ACMKDD International Conference on Knowledge Discovery and Data Mining*, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 283–291, dec 2008. ISBN 9781605581934. doi: 10.1145/1401890.1401928. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008 ; Conference date: 24-08-2008 Through 27-08-2008.

D Graves. Understanding the promise and limits of automated fact-checking. Technical report, Reuters Institute, 2018.

U. Hahn and I. Mani. The challenges of automatic summarization. *Computer*, 33(11):29–36, 2000.

Sanda M. Harabagiu, Andrew Hickl, and V. Finley Lacatusu. Negation, contrast and contradiction in text processing. In *AAAI*, 2006.

Julia Hirschberg and Christopher Manning. Advances in natural language processing. *Science (New York, N.Y.)*, 349:261–266, 07 2015. doi: 10.1126/science.aaa8685.

Tuan-Anh Hoang, Khoi Duy Vo, and Wolfgang Nejdl. W2e: A worldwide-event benchmark dataset for topic detection and tracking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1847–1850, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3269309. URL https://doi.org/10.1145/3269206.3269309.

Vasant Honavar. *Symbolic Artificial Intelligence and Numeric Artificial Neural Networks: Towards a Resolution of the Dichotomy*, pages 351–388. Springer US, Boston, MA, 1995. ISBN 978-0-585-29599-2. doi: 10.1007/978-0-585-29599-2_11. URL https://doi.org/10.1007/978-0-585-29599-2_11.

King Hussein. A study on nlp applications and ambiguity problems shaidah jusoh. 2018.

Shan Jiang and Christo Wilson. Linguistic signals under misinformation and fact-checking: Evidence from user comments on social media. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW), November 2018. doi: 10.1145/3274351. URL https://doi.org/10.1145/3274351.

Kayla N. Jordan, Joanna Sterling, James W. Pennebaker, and Ryan L. Boyd. Examining long-term trends in politics and culture through language of political leaders and cultural institutions. *Proceedings of the National Academy of Sciences of the United States of America*, 116(9): 3476–3481, February 2019. ISSN 0027-8424. doi: 10.1073/pnas.1811987116.

S. G. Kanakaraddi and S. S. Nandyal. Survey on parts of speech tagger techniques. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–6, March 2018. doi: 10.1109/ICCTCT.2018.8550884.

Muhammad Taimoor Khan, Mehr Yahya Durrani, Armughan Ali, Irum Inayat, Shehzad Khalid, and Kamran Habib Khan. Sentiment analysis and the complex natural language. *Complex Adaptive Systems Modeling*, 4:1–19, 2016.

Bushra Kidwai and Nadesh RK. Design and development of diagnostic chabot for supporting primary health care systems. *Procedia Computer Science*, 167:75 – 84, 2020. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2020.03.184. URL http://www.sciencedirect.com/science/article/pii/S1877050920306499. International Conference on Computational Intelligence and Data Science.

Julien Kloetzer, Stijn De Saeger, Kentaro Torisawa, Chikara Hashimoto, Jong-Hoon Oh, Motoki Sano, and Kiyonori Ohtake. Two-stage method for large-scale acquisition of contradiction pattern pairs using entailment. In *EMNLP*, 2013.

Piroska Lendvai, Isabelle Augenstein, Kalina Bontcheva, and Thierry Declerck. Monolingual social media datasets for detecting contradiction and entailment. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

Luyang Li, Bing Qin, and Ting Liu. Contradiction detection with contradiction-specific word embedding. *Algorithms*, 10:59, 2017.

Cheng Wen Lin. The study of political language : A brief overview of recent research. 2012.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. Identifying synonyms among distributionally similar words. In *IJCAI*, 2003.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. Learning semantic word embeddings based on ordinal knowledge constraints. 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 1501–1511, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1145. URL https://www.aclweb.org/anthology/P15-1145.

Yufeng Liu, Hao Helen Zhang, and Yichao Wu. Hard or soft classification? large-margin unified machines. *Journal of the American Statistical Association*, 106(493):166–177, 2011. doi: 10.1198/jasa.2011.tm10319. URL https://doi.org/10.1198/jasa.2011.tm10319. PMID: 22162896.

Mateusz Malinowski and Mario Fritz. Tutorial on answering questions about images with deep learning, 2016.

William C. Mann and Sandra A. Thompson. *Rhetorical Structure Theory: Description and Construction of Text Structures*, pages 85–95. Springer Netherlands, Dordrecht, 1987. doi: 10.1007/978-94-009-3645-4_7. URL https://doi.org/10.1007/978-94-009-3645-4_7.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval@COLING*, 2014.

David M. Markowitz and Jeffrey T. Hancock. Linguistic traces of a scientific fraud: The case of diederik stapel. *PLOS ONE*, 9(8):1–5, 08 2014. doi: 10.1371/journal.pone.0105937. URL https://doi.org/10.1371/journal.pone.0105937.

Marie-Catherine De Marneffe, Anna N. Rafferty, and Christopher D. Manning. Finding contradictions in text. In *ACL*, 2008.

Rada Mihalcea and Carlo Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, page 309–312, USA, 2009. Association for Computational Linguistics.

Lilyana Mihalkova, Tuyen Huynh, and Raymond J. Mooney. Mapping and revising markov logic networks for transfer learning. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, page 608–614. AAAI Press, 2007. ISBN 9781577353232.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

Amit Mishra and Sanjay Kumar Jain. A survey on question answering systems with classification. *Journal of King Saud University - Computer and Information Sciences*, 28(3):345 – 361, 2016. ISSN 1319-1578. doi: https://doi.org/10.1016/j.jksuci.2014.10.007. URL http://www.sciencedirect.com/science/article/pii/S1319157815000890.

Nikola Mrkšić, Diarmuid Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. pages 142–148, 03 2016. doi: 10.18653/v1/N16-1018.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. URL http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002. Publisher: John Benjamins Publishing Company.

Sakari Nieminen and Lauri Rapeli. Fighting misperceptions and doubting journalists' objectivity: A review of fact-checking literature. *Political Studies Review*, 17(3):296–309, 2019. doi: 10.1177/1478929918786852. URL https://doi.org/10.1177/1478929918786852.

Xiaoye Ouyang, Shudong Chen, Hua Zhao, Yuexing Hao, Wei Li, Shaojie Li, Rong Wang, and Xiaohu Liang. A multi-cross matching network for chinese named entity linking in short text. *Journal of Physics: Conference Series*, 1325:012069, oct 2019. doi: 10.1088/1742-6596/1325/1/012069. URL https://doi.org/10.1088%2F1742-6596%2F1325%2F1%2F012069.

P. Padmavathy, S. Pakkir Mohideen, and Zameer Gulzar. A novel architecture for a two-pass opinion mining classifier. In Raghavendra Rao Chillarige, Salvatore Distefano, and Sandeep Singh Rawat, editors, *Advances in Computational Intelligence and Informatics*, pages 27–35, Singapore, 2020. Springer Singapore. ISBN 978-981-15-3338-9.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

Andreas Peldszus and Manfred Stede. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon 2015 / Vol. 2*, pages 801–815, London, 2016. College Publications.

James W. Pennebaker, Linda E. Francis, and Roger John Booth. Liwc: Linguistic inquiry and word count. 2001.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke 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)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Adam L. Putnam, Christopher N. Wahlheim, and Larry L. Jacoby. Memory for flip-flopping: Detection and recollection of political contradictions. *Memory & Cognition*, 42(7):1198–1210, Oct 2014. ISSN 1532-5946. doi: 10.3758/s13421-014-0419-9. URL https://doi.org/10.3758/s13421-014-0419-9.

Alec Radford. Improving language understanding by generative pre-training. In *arxiv*, 2018.

Matthew A. Lambon Ralph. Distributed versus localist representations: Evidence from a study of item consistency in a case of classical anomia. *Brain and Language*, 64(3):339–360, 1998. doi: 10.1006/brln.1998.1976. URL http://pdfs.semanticscholar.org/4cda/3fa7a181720f64949880cf5d234ecc364729.pdf.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1317. URL https://www.aclweb.org/anthology/D17-1317.

P. Rayson. From key words to key semantic domains. *International Journal of Corpus Linguistics*, 13(4):519–549, 2008. ISSN 1384-6655. doi: 10.1075/ijcl.13.4.06ray. This article has been accepted for publication in International Journal of Corpus Linguistics, Volume 13, Issue 4, 2008, pages: 519-549, © 2008 John Benjamins, the publisher should be contacted for permission to re-use the material in any form.

Alan Ritter, Stephen Soderland, Doug Downey, and Oren Etzioni. It's a contradiction - no, it's not: A case study using functional relations. In *EMNLP*, 2008.

Asim Roy. The theory of localist representation and of a purely abstract cognitive system: The evidence from cortical columns, category cells, and multisensory neurons. *Frontiers in Psychology*, 8, 2017.

Victoria L. Rubin and Tatiana Vashchilko. Identification of truth and deception in text: Application of vector space model to rhetorical structure theory. 2012.

Edouard Ngor Sarr and Ousmane Sall. Automation of fact-checking: State of the art, obstacles and perspectives. *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 1314–1317, 2017.

Jacques Savoy. Trump's and clinton's style and rhetoric during the 2016 presidential election. *Journal of Quantitative Linguistics*, 25:168–189, 2018.

Roy Schwartz, Roi Reichart, and Ari Rappoport. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 258–267, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/K15-1026. URL https://www.aclweb.org/anthology/K15-1026.

Cheng-Wei Shih, Chengwei Lee, Richard Tzong-Han Tsai, and Wen-Lian Hsu. Validating contradiction in texts using online co-mention pattern checking. *ACM Trans. Asian Lang. Inf. Process.*, 11:17:1–17:21, 2012.

Maria Skeppstedt, Andreas Peldszus, and Manfred Stede. More or less controlled elicitation of argumentative text: Enlarging a microtext corpus via crowdsourcing. In *Proceedings of the 5th Workshop on Argument Mining*, pages 155–163, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5218. URL https://www.aclweb.org/anthology/W18-5218.

Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659, September 2017. doi: 10.1162/COLI_a_00295. URL https://www.aclweb.org/anthology/J17-3005.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139 – 162, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2020.01.010. URL http://www.sciencedirect.com/science/article/pii/S1566253519303677.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *INTERSPEECH*, 2012.

Oguzhan Tas and Farzad Kiyani. A survey automatic text summarization. *PressAcademia Procedia*, 5:205 – 213, 2017. doi: 10.17261/Pressacademia.2017.591.

James Thorne and Andreas Vlachos. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1283.

Mireya Tovar, Gerardo Flores, José A. Reyes-Ortiz, and Meliza Contreras. Validation of semantic relation of synonymy in domain ontologies using lexico-syntactic patterns and acronyms. In José Francisco Martínez-Trinidad, Jesús Ariel Carrasco-Ochoa, José Arturo Olvera-López, and Sudeep Sarkar, editors, *Pattern Recognition*, pages 199–208, Cham, 2018. Springer International Publishing. ISBN 978-3-319-92198-3.

Mikalai Tsytsarau. Scalable detection of sentiment-based contradictions. 2011.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Jacobus Visser, Barbara Konat, Rory Duthie, Marcin Koszowy, Katarzyna Budzynska, and Chris Reed. Argumentation in the 2016 us presidential elections: Annotated corpora of television debates and social media reaction. *Language Resources and Evaluation*, 54:123–154, March 2019. ISSN 1574-020X. doi: 10.1007/s10579-019-09446-8.

Svitlana Volkova, Kyle J. Shaffer, Jin Yea Jang, and Nathan O. Hodas. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. 7 2017. doi: 10.18653/v1/P17-2102.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

Yuqi Yu, Guannan Liu, Hanbing Yan, Hong Li, and Hongchao Guan. Attention-based bi-lstm model for anomalous http traffic detection. *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–6, 2018.