FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Dissecting Fact-Checking Systems: The Impact of Evidence Extraction Methods

Pedro José Lourenço Azevedo

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Doctor Henrique Daniel de Avelar Lopes Cardoso

Second Supervisor: Gil Filipe da Rocha

Third Supervisor: Doctor Diego Nascimento Esteves da Silva

July 30, 2020

Dissecting Fact-Checking Systems: The Impact of Evidence Extraction Methods

Pedro José Lourenço Azevedo

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor João Pedro Carvalho Leal Mendes Moreira
External Examiner: Doctor Ricardo Daniel Santos Faro Marques Ribeiro
Supervisor: Doctor Henrique Daniel de Avelar Lopes Cardoso

July 30, 2020

Abstract

In today's society, with global access to the internet and online posting, we are provided with new information about any subject or topic on a daily basis. With free access on a global scale, it is easy for anyone to share virtually anything, giving raise to what is commonly identified as *fake news*. Fake news are typically shared with a bad intent, where although some people see them as true, they are actually based on unsustained claims.

Given how pervasive these issues are, automated systems can support users in processing such amounts of information and verify different sets of information. Fact-checking is an example of such systems, based on algorithms and common-sense rules, which aims to determine the truth value of claims.

The topic of fact-checking has captured much attention not only in media coverage, but also in many fields of research, such as natural language processing, knowledge extraction and aggregation, and computational journalism. From these initial efforts, several challenges have been proposed, as the Fact Extraction and Verification (FEVER) challenge is one such example. FEVER challenged teams to create a system which could retrieve evidence in order to make an assessment of a given claim based on Wikipedia documents.

This thesis explores the FEVER dataset and contributes to improving the DeFactoNLP system, one of the system participant on the FEVER challenge. This system is divided in three main task: Document Retrieval, Evidence Retrieval, and Label Classification. One ablation study of the system was conducted to evaluate the performance of each component in isolation as well as to assess the propagation of errors in the pipeline of tasks proposed in the DeFactoNLP system.

For the Document Retrieval task, it was found that a NER technique was not retrieving any document for 10% of the claims, and that improving the recall would impact the overall performance of the system. To address these issues, we used the Open Information Extraction technique with external data sources, and achieved state-of-the-art results. For the Evidence Retrieval task, our analysis show that a better precision and a fixed number of retrieved sentences have the potential to improve the overall system. We then propose a Triple-base model and explore two state-of-the-art approaches that leverage recent language models and Transformer-based architectures, using either a pairwise or pointwise methods. The results presented here show that the pointwise architecture performs the best, significantly improving the DeFactoNLP system. Through an error analysis, it was found that the presence of non-solved anaphoric references were negatively impacting the models in the Evidence Retrieval task. To tackle this issue, a new Wikipedia dump was created using coreference resolution state-of-the-art models to solve the corresponding anaphoric references. Training the proposed models in this dump had a positive impact, since all the models increased their performance.

Keywords: Fact Extraction, Fact Verification, Open Information Extraction, Natural Language Processing, Deep Learning, Artificial Intelligence, Information Retrieval

Resumo

Na sociedade de hoje, com acesso global à internet e à afixação online, é-nos fornecida diariamente nova informação sobre todos os assuntos ou tópicos. Com acesso gratuito a nível global, qualquer pessoa pode facilmente partilhar algo de forma virtual, aumentando a importância de um problema comummente apelidado de *fake news* (notícias falsas, em tradução livre). Estas notícias falsas são maioritariamente divulgadas com alegadas más intenções e, embora sejam encaradas por alguns como verdade, não são baseadas em factos reais.

Sendo esta questão muito preocupante, os sistemas automáticos podem ajudar a processar e filtrar informações, verificando e comparando-as com outros conjuntos de informações. *Fact-checking* é um exemplo de um dos referidos sistemas, baseado em algoritmos e regras de senso comum, classificando uma afirmação como fidedigna ou não fidedigna.

O termo *fact-ckecking* tem captado muita atenção não só dos media, mas também de diversos campos de pesquisa, como o processamento de linguagem natural, extração e agregação de conhecimentos e jornalismo computacional. Desde o início que foram então propostas várias competições, sendo a mais emblemática a Extração e Verificação de Factos (FEVER). A FEVER desafiou equipas a criar um sistema que fosse capaz de retirar evidências, de forma a avaliar uma determinada afirmação utilizando documentos do Wikipedia.

Esta dissertação explora o conjunto de dados FEVER e contribui para melhorar o sistema DeFactoNLP, sendo este um dos sistemas que participaram na competição FEVER. Este sistema está dividido em três componentes principais: Extração de Documentos, Extração de Evidências e Classificação da Afirmação. Foi conduzido um estudo minucioso do sistema DeFactoNLP, para avaliar o desempenho de cada componente, bem como da propagação de erros pelas mesmas.

Em relação à componente de Extração de Documentos, verificou-se que a técnica NER não estava a extrair nenhum documento para 10% das afirmações e que melhorar o *recall* iria ter um impacto positivo no desempenho global do sistema. Para abordar estas questões, foi utilizada a técnica *Open Information Extraction* aliada a fontes de dados externas, obtendo resultados estado-de-arte. Na componente Extração de Evidências, a nossa análise demonstra que uma melhor precisão e um número fixo de frases recuperadas possibilitam melhorar o sistema global. Foi proposto um modelo baseado em Triplos e exploradas duas abordagens estados-de-arte que aproveitam modelos linguísticos recentes e arquiteturas baseadas em *Transformers*, utilizando operações *pairwise* ou *pointwise*. Os resultados comprovam que a arquitetura tem o melhor desempenho, melhorando significativamente o sistema DeFactoNLP. Através de uma análise de erro, verificou-se que a presença de referências anafóricas não resolvidas estava a ter um impacto negativo nos modelos na tarefa de Extração de Evidências. Por conseguinte, foi criado um novo *Wikipedia dump* usando resolução de coreferências para resolver as referências anafóricas. Este novo *Wikipedia dump* teve

um impacto positivo, uma vez que todos os modelos aumentaram o seu desempenho.

Palavras chave: Extração de Factos, Verificação de Factos, *Open Information Extraction*, Extração da Informação, Processamento da Linguagem Natural, Aprendizagem Profunda, Inteligência Artificial

Acknowledgements

Thanks to my family for all the support given.

I also want to thank my supervisors for their constant feedback and availability throughout the development of this entire document.

A big thank you to all the people who have always stood by my side and helped me during this process by actions, words or simply for being there.

Pedro Azevedo

"Every man has his secret sorrows which the world knows not,
and often times we call a man cold when he is only sad."

Henry Wadsworth Longfellow

"The best fights are the ones in which you don't need to look for your strength.
It's just there"

Pedro José Lourenço Azevedo

Contents

List of Figures

List of Tables

Abbreviations

FEVER	Fact Extraction and VERification
TF-IDF	Term Frequency-Inverse Document Frequency
NER	Name Entity Recognition
OIE	Open Information Extraction
RF	Random Forest
TB	Triple-Based
RTE	Recognizing Textual Entailment
NLP	Natural Language Processing
API	Application Programming Interface
AM	Argumentation Mining
HMM	Hidden Markov Model
IR	Information Retrieval
POS	Part-Of-Speech
QA	Question Answering
CR	Coreference Resolution
CRF	Conditional Random Fields

Chapter 1

Introduction

In different research areas, a *fact* may have distinctive meanings. In philosophy, facts may be perceived as information that one can use to prove a sentence as a true sentence [49]. In law, facts are the information in which lawyers base their arguments, and those facts are, in turn, based on evidence, which has been already proven [105]. In science, a fact is something extracted based on experiments. In all of these areas, there is something in common: there must be something to prove a particular fact, not making any prompt assessment on some perspective or conclusion.

In people's daily lives, the ability to include facts into their speech is something that cannot be neglected. An important question arises: *Are those facts true?* This question can only be answered by presenting at least one evidence. Using proven facts can make a speech much stronger, whether in argumentative texts or debates, in a political context or otherwise. Being able to disprove a person's statement, based upon specific evidence is a powerful weapon to win argumentative situations as the ones illustrated before. In this work, a fact is considered a statement or assertion of verified information about something that exists or has occurred. This must be followed by an evidence that can either prove or disprove the fact.

Internet browsing is becoming a common activity in most people's every-day life. Web search usage has been increasing more and more throughout time, with keyword-based access to find documents providing information at all levels. The most used website is Google.com with billions of monthly visits, next comes YouTube.com close to 24 billion, and Facebook.com with almost 20 billion [99]. Regarding the use of Facebook and other social networks, a person that writes any kind of information can reach millions of people within hours [2]. In 2016, Google and Facebook employed models preventing a spread of fake news in their advertising services showing only true news and information [120]. Although this was promising news, Google cannot ensure that the information on the documents provided is all true, because of the lack of mechanisms to do so. Also, the information provided for each of Facebook's users is chosen mainly by the user because the presented information is based on their user friends and Facebook is being used more and more as a source of information [77].

However, any piece of information may or may not be true, and this can cause a change in people's attitudes, including voting intentions [40]. Adding to a recent study which finds that false news spread more quickly than true ones [115], the ability to verify the truthfulness of a fact (e.g., by finding evidence from reliable sources) becomes essential in any public website. In the journalistic field, the number of requests to automate this process is very significant [22, 4, 41].

1.1 Motivation

When a statement is established forward as a fact, there is always the question of its veracity. To aggravate the situation, there is rarely information or sources proving the alleged fact or how it was inferred. Checking a statement manually may take from hours to a few days depending on the complexity of its facts [46]. This process is usually carried out by professionals who, at a first instance, will have to understand the statement and verify its source and what it refers to. Then, they will have to look for previous facts, such as analyzing whole speeches, debates, legislation, among others, in order to extract as much information as possible to form evidence. Only after this process is concluded, it is possible to make an assessment of the statement and turn it into a proven fact.

The manual work of fact-checking is very exhaustive. As for its automation, it is still necessary to overcome several challenges that are explained by Sarr et al. [95]. Some of these challenges intersect with those in some specific aspects of Natural Language Processing (NLP), a field addressing the ability of a machine to read and process text:

- **The subjectivity of reliability:** Reliability or trust is something that is built over time. Assessing information source over time depends on who assessed the facts before. It also remains a question of whether source consistency of reliability is maintained;
- **Unstructured data:** All information is presented differently. However, it is known that unstructured data contains more information than structured data. With this, it is necessary to create a robust mechanism for the extraction of information, wherever it is and however it may appear;
- **The temporal aspect of the reliability and the lack of sources:** When time is considered, checking a fact becomes even more problematic because of having to know when something was, is or will be true. Not knowing the precedent and origin of the source of information, puts a question mark over the reliability of the fact;
- **The semantics of sentences:** People who write texts never write it in the same way. Often sentences are not written with the right structure. Despite the advances of NLP in this area, it cannot fill and detect all the variations and imperfections that a human may have. Therefore, to check something, since it is necessary to extract information from different sources, the format of the text, the structure of a sentence, and numbers of errors of the information will appear in different ways. This will complicate the whole process of fact-checking;

- **The identification of factual facts:** A factual fact will always be true, therefore timeless. Its use should be a priority, but it is not always of interest. Evaluating a sentence as a factual fact can create an increase in optimization in future iterations as a better identification of important facts, since there is no need for to check the temporal aspect of the fact;
- **The real time and speed of information spreading:** The information continues to grow exponentially. For a human fact-checker, keeping up with this constant overwhelming flow of information can make a person's work obsolete and irrelevant [47]. The speed with which news and information are passed between people, for example, due to the use of social networks, there is an increase of interest for users to be able to check, in real time, the information presented;
- **The format in Edition:** An engaging source of information can be the articles of public opinion. Newspapers publish many argumentative texts, on the Internet a lot of texts are published and presented. However, none of them follow a basic rule of information, also called meta-data. The information presented generally displays only three fields: the author, the title and the text. However, other types of information are also very important, such as the date of publication, information regarding the author, what sources the author used, among others;
- **Relation between evidence:** For a statement to be verified, just one piece of evidence is often not enough, proving necessary a whole set of evidence to verify the statement. This presents a challenge because it may be necessary to connect several pieces of evidence and determine which may depend on another and how they relate to a statement.

These challenges are all authentic, and the need to overcome them is increasing in interest.

One phenomenon that has seen an increase, in social networks, is *fake news*. This arises from the growth in the amount of information that is passed on. With information also comes misinformation (the creation of rumours) [73], astroturfing (the veiling of a message or movement sponsored by a company or enterprise, but leading people into thinking it was a community effort) [87], spam (referred to as electronic junk) [24] and outright fraud (very similar to phishing by impersonating a trustworthy third party) [52]. In this way, fact extraction and fact verification areas are growing, and more than ever a demand from different business areas and enterprises arises becoming a perspicuous motivation to surpass the main fact-checking challenges.

1.2 Objectives

The main objective addressed in this dissertation is **to create an ablation study of a fact-checking system and, by identifying the flaws, to improve the system using different Evidence Extraction methods.** The technical goal is to identify at least one evidence, allowing the assessment of a given claim by retrieving information from unstructured data. Therefore, if an evidence is found to support a claim, the claim will become a fact.

Fact-Checking systems currently consist of three main steps and each step has his own objective:

- **Document Retrieval** – through a given statement, lies the ability to find documents that are likely to contain information relevant to the fact. Improve this step using a complete semantic and syntactic analysis of the fact and enabling a computing relation with relevant retrieved documents.
- **Evidence Retrieval** – through an analysis of the documents content, lies the ability to retrieve possible evidences for the fact. Improve this step by creating a rank system to evaluate how fitting is the evidence to be able to verify a statement.
- **Label Classification** – through the statement and the extracted evidence, verify the veracity of the fact. Create metrics to evaluate the overall system being able to verify, besides the label given and the evidence retrieved, if the label of the fact is based on the provided evidence.

Each of these three steps are treated independently, allowing a separation. By doing so, the evaluation mechanisms for each of the phases undertaken will be addressed in this dissertation. This stems from the fact that, for the most part, evaluation methods only exist at the end of a system. Yoneda et al. [127] noticed that retrieving evidence that can refute a fact is harder than retrieving a supporting one. This happens due to the lack of similarity between facts and evidence. Therefore, a separate assessment for different types of facts should be considered carefully to always have a balance between refutable and verifiable facts.

A well-known issue of pipeline architectures is the propagation of errors. Reporting scores on the final outcome of the system without studying the performance of underlying components may hide problems in earlier stages, thus potentially degrading the overall performance. Therefore, the fundamental research questions explored in this dissertation are as follows:

- How each fact-checking system component can impact the overall performance?
- Can Open Information Extraction methods be used to help retrieve documents and possible evidence?
- Are transformer-based models, the state-of-the-art in NLP, able to outperform Open Information extraction methods?

1.3 Scientific Contributions

In this dissertation, one of the proposed systems – *DeFactoNLP* [88], which achieved the fifth place in Evidence F1 Score at the FEVER 1.0 Shared Task[1] – is dissected, and an ablation study

[1] https://fever.ai/2018/task.html

is performed to obtain concrete insights on the impact that each component has in the overall performance. Based on the ablation study, the *Document Retrieval* and *Evidence Retrieval* stages are improved by re-designing them using and applying *Evidence Extraction Methods*, such as Open Information Extraction and transformer-based models. Therefore, the main scientific contributions of this dissertation are:

- An **Ablation Study** on the importance of the different metrics in the different stages of the system was assessed.
- In the **Document Retrieval** task, after a thorough search of the current literature, the state-of-the-art is achieved.
- In the **Evidence Retrieval** task, is provided how different approaches can help improving evidence retrieval.
- A new **dataset dump** based on the provided dataset [110], which improves the performance of the models.

1.4 Document Structure

The remainder of this dissertation is structured as follows:

In chapter 2 a background of the main topics in Natural Language Processing are presented related to this dissertation.

In chapter 3 the main stages of a fact checking system are described, making the connection between different research areas and the datasets that can evaluate the developed systems on fact-checking. In the end, the core systems are described in detail.

Chapter 4 dissects the DeFactoNLP system, evaluating every stage. The chapter is divided into three sections, each one representing a task: Document Retrieval (Section 4.1), Evidence Retrieval (Section 4.2), and Label Classification (Section 4.3).

Chapter 5 includes the first system improvement, where the Document Retrieval task is re-designed. This chapter is divided into four sections, representing different approaches, each one motivated by an error analysis or some limitation of the previous approaches.

Chapter 6 includes the second system improvement, where the Evidence Retrieval task is re-designed. Different models are explored, and the impact on those models in the end-to-end system is shown.

Chapter 7 points the main conclusions withdrawn from this work and pinpoints some lines of future work.

Chapter 2

Background

Natural Language Processing, also known as *NLP*, is a branch of Artificial Intelligence that attempts to unify, communication, either written or spoken, between the computer and the human through computational linguistic methods. Thus, a machine tries to learn and manipulate a text or speech [18][23] for a given purpose. Many applications are already based on NLP, for example, to make a summary of a text [34], the construction of generic or domain-specific chatbots, structuring of argumentative texts [92], question answering, among many other applications.

In this chapter, the background of the most relevant topics for this dissertation will be presented. Each section will start to give a definition of the topic and a brief history of it and, finally, a brief analysis of the topics strengths and applications. Section 2.1 starts with a brief historical context focusing on the most significant applications in the Natural Language Processing (NLP) field. All of those applications, including the most recent ones, usually follow several text transformations as explained in Section 2.2. Section 2.3 focus on how to represent words in a certain vector space, and how such methods can be combined to represent sentences. Then, this chapter focuses on more specific tasks such as Open Information Retrieval (OIE) and Coreference Resolution (CR). In Section 2.4, OIE is presented with an analysis of the relevant literature by explaining each system, comparing their performances, and highlighting the positive aspects. In Section 2.5, the theoretical aspect of the Coreference Resolution is explored. A thorough analysis of the relevant literature on Fact-Checking systems revealed that, to the best of our knowledge, Coreference Resolution has never been used. However, in this dissertation, this technique is employed as an attempt to improve the performance of fact-checking, more specifically in the *Evidence Extraction* and *Label Classification* stages.

2.1 NLP before Data-Driven Models

The first effort for a machine to be able to interpret text began around the 1960s. The two most significant systems were designed to bring about a revolution in the area: ELIZA (1966) [118] and

SHRDLU (1971) [121].

ELIZA was developed at MIT Artificial Intelligence Laboratory by Joseph Weizenbaum. The main goal was the creation of a conversation between a machine and a human. The conversation is carried out by the recognition of specific keywords. Through them, a set of questions are formulated that were already defined. One of the most positive aspects of this system was the beginning of keyword recognition, context recognition by changing sentences to include keywords or even if keywords are not identified, the ability to change sentences to be relevant in the context of the conversation.

SHRDLU was a system, also developed in MIT, that supported commands obtained from a conversation with the user. Those commands would be executed in a world with objects, making them move, giving names to the objects, and even reporting the state of the world. Through the use of adjectives, the system can identify different objects. Through the use of verbs, the system can recognize actions. Another fundamental aspect was also a memory system for the system to know the last actions and know names given to objects to simplify conversations and identification.

These two programs have created a wave of optimism for the creation of new solutions to retrieve information from text. However, their application in the real-world has not been very successful due to the ambiguity of natural language. Since then, an area of research called generative grammars has begun to grow. This area focuses on the creation of rules that allow all the possibilities of correctly constructing a sentence. Hence, Chomsky proposed a new way of describing natural language called Transformational Grammar, where each sentence can be represented in two aspects: a deep structure representing all the semantic relations, and a surface structure that can be expressed with transformations [16, 15].

Natural language began to be represented by rules deeply explored throughout the 1980s. Thus a need was created to find a way to structure information, that is, structured knowledge bases, namely ontologies proposed by Gruber in 1991 [43]. Ontologies contain information about things from a given domain, organized in taxonomies, and linked together through properties [113]. This line of research was explored during the 1990s, enabling the creation of Knowledge Graphs that can be described as large ontologies. However, other contributors have pointed out that knowledge graphs are somehow superior to ontologies and provide additional features [30].

All of this enabled machine learning algorithms to influence Natural Language Processing shifting the focus of NLP tasks to reach data-driven solutions. Collobert et al. [23] presented one of the first works using deep learning, creating an attempt to generalize the four main tasks of NLP: part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. It used a neural network architecture, achieving baselines for those tasks.

2.2 Basic Pipeline in NLP

When analyzing sentences, it is necessary to develop mechanisms that enable a machine to understand them. Furthermore, a machine also needs to analyze the different words and the relationships between them. There is no single or general pipeline since it will always be different to fulfill the

primary goal. However, over time, numerous techniques have been developed to process the texts presented efficiently. The main objective of these techniques, when placed sequentially, is the generation of a text with less noise or, even more important, the ability to create features for the different models of Machine Learning. Thus, this section will present the most common steps in a pipeline to perform a successful text treatment and an analysis of the sentences in a text such as tokenization, lemmatization and stemming, part-of-speech tagging and parsing trees.

2.2.1 Tokenization

The first step that almost all pipelines begin with is a process of tokenization. This process refers to the ability to, given a text, separate it into different words, that have come to be called tokens. If a *space* is considered as a word boundary, this task may seem trivial. Although several exceptions can occur, these same exceptions can be different and specific for each language. Following, some of those exceptions are stated:

- Some nouns, if separated into different tokens, will have no meaning or will just be redundant. For example, the city *Los Angeles* would be separated into two different tokens, but it is convenient not to do it. The reason is not to lose the meaning of the name of the city.
- There are words that are not always separated by space, but by *hyphens*, being called compound adjectives or nouns. The word *off-state* can also be written without the hyphen being the separation of sentences by spaces not enough. In Germany, compound nouns are written in the same word without any separation, becoming a challenging task where and when to divide them.
- In a text, a *period* generally means the end of the sentence. It is the same as a sentence boundary. However, the period can also be used for abbreviations or acronyms, so it should not be separated into different tokens.

In this regard, all the main NLP libraries have their tokenizer module or use specific libraries whose only goal is to tokenize. A set of regular expressions is the primary way that those libraries create tokens. Stanford Penn Treebank Tokenizer, from The Stanford CoreNLP library [70], has a slightly different approach. It uses a finite automaton, with heuristics to decide when and where to split.

2.2.2 Lemmatization and Stemming

While words may have the same meaning, others may belong to the same word family. This happens because one word may contain different forms (work, works, working). In some specific areas, such as Information Retrieval, the probability of finding documents decreases with the increasing of vocabulary. In general, by removing the suffixes, keeps the essential information from words. On the other hand, prefixes are not common to be removed because they may represent the denial function. Altering the words in a text can make a crucial step in an NLP pipeline. So,

if different words contain the same word, it is necessary to find ways to reduce the vocabulary by modifying words that belong to the same word family or share the same root. This way, there are two main processes to deal with this: lemmatization and stemming.

According to *Oxford University Press*[1] (OUP), a lemma is "a heading indicating the subject or argument of a literary composition, an annotation, or a dictionary entry.". Transforming words to their lemma will induce removal of all suffixes and prefixes and even the complete modification of certain words (better -> good). For this process to be possible, it is necessary to create word pairs that can create these same relationships: (inflicted form, lemma).

Stemming does not always originate valid words, so unlike lemmatization, it does not need a prior language vocabulary, although the rules are different from language to language. Through the creation of rules, the central concept is to be able to transform the suffixes and prefixes of the word into something different. The first known stemming algorithm was created by Lovins [66], having an extensive set of rules. However, this was primarily surpassed by Porters stemmer [84], which is the most commonly used, and Paice's algorithm [80].

Sometimes, Part of Speech Tagging is used before lemmatization because it can alter the process by knowing the part of speech of the word.

2.2.3 Part of Speech Tagging

Part of speech (POS) tagging and named entity recognition can be complemented to perform specific tagging for each word [76]. However, the recognition of entities by specifying their type (person, country, date) does not help as much as POS for the steps of different pipelines. Part of speech can be used as a feature by selecting the word senses from WordNet [75]. However, a different tagging may depend on the corpus or language used. The two most commonly used corpora for English are the Brown corpus [36] and the Penn Treebank [71]. The latter is the most common. An overview of the tags used by the Penn Treebank is presented in Table 2.1.

Hidden Markov Model (HMM) [6] is a solution for performing POS tagging. This solution is based on a simplification of a Bayesian network where each observed state has a corresponding hidden state depending only on the previous hidden state.

By creating a simple POS tagging model, it is possible to achieve 90% accuracy [10]. This accuracy is possible because most English words have only one possible match. This value is reached only by analyzing a corpus and, for words that contain more than one possible tag, the most common is selected. For unknown words, the proper noun is used. 90% can seem a very high value; however, if only considering words that may have more than one tag, the value would be lower. Using Hidden Markov models as a base, the POS tagging system obtains about 95% accuracy [97]. With this result, a Second Order HMM was proposed reaching 96% accuracy making use of more contextual information than standard statistical systems for the approximations used in the second-order [108].

[1] Lexico.com

Table 2.1: Tags proposed in the Penn Treebank [71]

Tag	Description	Example	Tag	Description	Example
CC	coord. conjunction	*and, or*	**RB**	adverb	*extremely*
CD	cardinal number	*one, two*	**RBR**	adverb, comparative	*never*
DT	determiner	*a, the*	**RBS**	adverb, superlative	*fastest*
EX	existential there	*there*	**RP**	particle	*up, o*
FW	foreign word	*noire*	**SYM**	symbol	+, %
IN	preposition or subconjunction	*of, in*	**TO**	"to"	*to*
JJ	adjective	*small*	**UH**	interjection	*oops, oh*
JJR	adject., comparative	*smaller*	**VB**	verb, base form	*fly*
JJS	adject., superlative	*smallest*	**VBD**	verb, past tense	*flew*
LS	list item marker	*1, one*	**VBG**	verb, gerund	*flying*
MD	modal	*can, could*	**VBN**	verb, past participle	*flown*
NN	noun, singular or mass	*dog*	**VBP**	verb, non-3sg pres	*fly*
NNS	noun, plural	*dogs*	**VBZ**	verb, 3sg pres	*flies*
NNP	proper noun, sing.	*London*	**WDT**	wh-determiner	*which, that*
NNPS	proper noun, plural	*Azores*	**WP**	wh-pronoun	*who, what*
PDT	predeterminer	*both, lot of*	**WP$**	possessive wh-	*whose*
POS	possessive ending	*'s*	**WRB**	wh-adverb	*where, how*
PRP	personal pronoun	*he, she*			

With the appearance of deep learning techniques, by using state of the art models, it was possible to achieve 97% accuracy with Conditional Random Fields (CRF) with structure regularization [107]. By creating an ensemble between CRF and a bidirectional Long Short Term Memory (BI-LSTM) the model achieved the state of the art results in POS Tagging [51].

2.2.4 Parsing Trees

All the steps described so far analyze natural language without worrying about the order in which words appear in a sentence. However, understanding the relationship between words is a task in which syntactic parsing focuses on. This task creates word links through the generation of trees. It is a mediation between linguistic expression and meaning.

Constituency trees subdivide a sentence into several sub-sentences. Similar to a binary tree, a node can be subdivided up to two nodes. Generally, the first nodes of the tree consist of a noun phrase (NP), a verbal phrase (VP), and a prepositional phrase (PP). The leaves will contain the words of the sentence and its part of speech tag. Thus, it is possible to state that this type of tree groups different words that together have a meaning in the sentence. In Fig 2.1, an example of a constituency tree is shown, for the phrase *He eats spaghetti with meat*. It is possible to infer, with a depth first search, that VP *eats* relates the NP *He* and the NP *spaghetti*.

Unlike constituency trees, dependency trees may not relate all the words. However, relationships are created where the nodes are the words, and the edges represent those relations that indicate direct links between words, labeled as syntactic relations. In Fig 2.2, an example of a dependency tree is shown, for the phrase *He has good control.*.

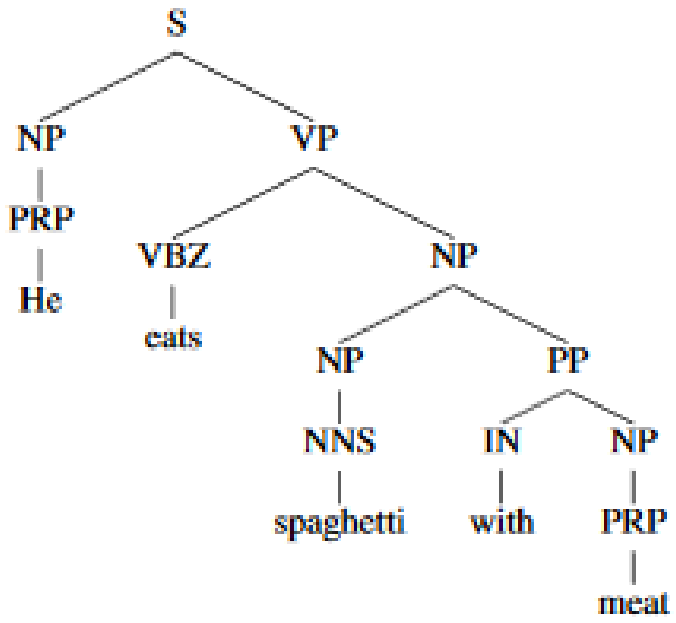

Figure 2.1: An example of a desired constituency tree [101]

2.3 Semantic Models and Word Embeddings

A machine learning model needs features, and these same features have to be, mainly, in numerical values, and the first need arises: the ability to transform the text into numbers to function as input for the models.

One of the most known transformation methods is Bag-of-Words (BoW). This transformation needs, in a first instance, a vocabulary and, through it, count the frequency of each word. Although it is simple, it can undoubtedly get better results through the lemmatization of words. This happens since BoW generates sparse vectors due to a large vocabulary. Lemmatization reduces the vocabulary size, hence, the BoW model generates more concise vectors. Another way to improve is to reduce vocabulary by removing stop-words or even removing words with almost one or a non-substantial frequency. Another very well known transformation is called Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF allows normalizing the frequency of words in a whole corpus [69].

Although these are reasonable solutions to relatively simple problems, in the area of Information Extraction, the need to find similar phrases or words is imperative. For example, the sentence "A student got great grades" can also be written as "A graduate obtained good classifications". Analyzing the words by their frequency will indicate a nearly zero match due to the lack of common words. In this way, it should be possible to present words in the form of vectors to capture semantic relationships. Representing words or texts as numerical vectors is called word embeddings. Thus, some unsupervised methods, to represent documents in a certain vector space, are

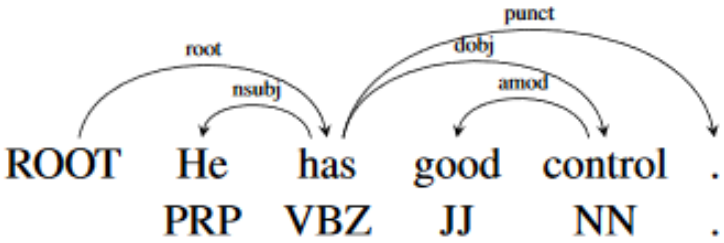

Figure 2.2: An example of a desired dependency tree [12]

based on word frequency in documents, but especially on their co-occurrence:

- **Latent Semantic Analysis** (LSA) [59] is a method based on the construction of matrices for each document in a certain corpus. These matrices count the number of occurrences of certain words. A mathematical method called singular value decomposition (SVD) is used to reduce the number of rows. LSA also assumes that if similar pieces of text contain the same words, those words have a close meaning. By performing this matrix construction, it is possible to evaluate how similar the documents are.

- **Latent Dirichlet Allocation** (LDA) [7] is based on the explanation of the similarity between documents through latent topics. Thus, a generative probabilistic model will observe documents finding relationships between words to create these latent topics to represent the information in the best way.

- **Word2vec** [74] is based on a feed-forward neural network capable of training word embeddings. The result of this model is the probabilistic prediction of a word appearing in a given context window. It is one of the most popular models when creating relationships between similar words and, through each word being represented in a multidimensional space, can find similar semantic value. For example, the words *king* and *queen* would be at the same distance as *men* and *women*.

- **GloVe** [82] differentiates of the word2vec model by creating a global co-occurrence matrix. GloVe embeddings are achieved by estimating, given a word, the probability if it will co-occur with other words. This presence of global information makes GloVe embeddings ideal for tasks such as word analogy, word similarity, and named entity recognition. However, the best benefit is that the training process is faster than word2vec. In order to compete with GloVe embeddings, a faster word2vec was created named FastText [8] that also can compute vectors for unknown vocabulary. This is possible since FastText word vector is built from subwords of words and also from substrings of characters. If the input contains misspelled words or even unknown words, this will still manage to build vectors.

- **ELMo** (Embeddings from Language Models) [83] generates the vectors for word representation using Recurrent Neural Networks (RNN), more specifically Long Short Term Memory (LSTM) units, rather than a feed-forward neural network such as used in Word2vec. LSTM has the main advantage of being able to solve dependencies efficiently. In Peters et al. [83] it is proposed to use bidirectional LSTM, using two LSTM: a forward one (left to right) and a backward one (right to left). These are concatenated at the end in the architecture generating the final embeddings. This type of architecture allowed us to start representing context-based sentences, thus increasing performance in major NLP tasks such as Question Answering, Co-reference Resolution, and Translation. ESIM model [14], enhanced sequential inference model, is based on LSTMs chains, carefully designed sequential inference model.

- **BERT** (Bidirectional Encoder Representations from Transformers) [27] was trained through the unlabelled data use of Wikipedia, containing about 2,500 million words and in a corpus of several books, containing about 800 million words. The BERT model is based on Sequence to Sequence models, consisting of two self-attention units: an Encoder and a Decoder. Unlike ELMo Embeddings, the contexts left to right and right to left are dependent on each other. In Figure 2.3, it is possible to notice the difference between BERT and ELMo model architectures, highlighting the contextualization differences. Looking at the GLUE Benchmark [116], BERT achieved state-of-the-art results in over 20 tasks. Several models were created based on BERT's architecture, such as AlBERT [58] and RoBERTa [65]. The pre-trained model is done by using a special token [MASK], which never occurs in the fine-tuning stage. This leads to one of the big problems of these models: a constant need for a fine-tuning for every specific task to achieve good results.

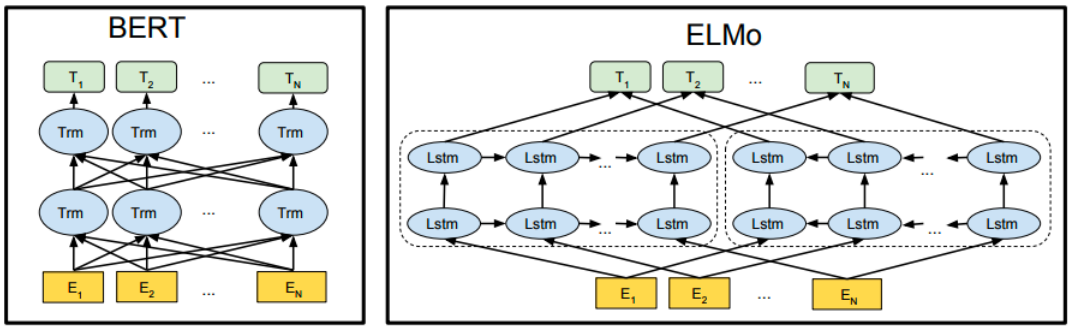

Figure 2.3: ELMo and BERT architectures [27]

2.4 Open Information Extraction

Open Information Extraction (OIE), as a research field, was firstly presented in 2007 by Banko et al. [5]. This research area aims to extract relations from a sentences creating tuples of information. Thus, information in natural language text can be represented in a structured way using verbal phrases and their arguments. In its purest form, given a natural language sentence, OIE techniques can extract information forming tuples in the form of a triple, consisting of subject (S), relation (R) and object (O).

Suppose we have the following input sentence:

AMD, which is based in the U.S., is a technology company.

It is possible to extract the two following triples:

("AMD"; "is based in"; "U.S.")
("AMD"; "is"; "technology company")

The formation of Triples can be achieved by specific algorithms using Dependency Parse trees [93, 54]. The general work of these algorithms, with a Dependency Parse Tree, through a depth first search, is to find nouns, verbs, and adjectives obtaining relations between them.

The first Open Information Extraction system using self-supervised learning was presented in 2007, called TextRunner [126]. ReVerb, an OIE software, introduced two new constraints: syntactic, that eliminates incoherent and uninformative extractions, and lexical, that separates valid relation phrases from an overly-specific relation [33]. OLLIE [96] was presented in 2012 based on ReVerb, making several improvements. The system's main features are the extraction of relations mediated by nouns, adjectives, and more by learning extraction patterns. ClaussIE [26], as the name suggests, proposes an approach through clauses. Thus, the sentences are divided into clauses, and the type of clause presented can be identified. OpenIE4 [72] ensembles all the previous techniques into one having as base the work develop in OLLIE. Using ClausIE as the base, MinIE's [38] main contribution was to mitigate overly-specific relations. This contribution has lead to a noise removal, achieving state-of-the-art scores. RnnOIE [104] used a BiLSTM and IOB (Inside–outside–beginning) tagging. Seq2seq OIE [25] used a sequence to sequence model and a copy mechanism achieves similar scores by using state-of-the-art models in Open IE task.

In Table 2.2 comparison values obtained by Zhan et el. [129] on the most recent OpenIE systems are presented. A benchmark was created for Open Information Extraction [103], where the scores are based on area Under the Receiver Operating Characteristic Curve (AUC) on the OIE2016 corpus.

[2] https://github.com/gabrielStanovsky/oie-benchmark

Table 2.2: Comparison between different systems by AUC on OIE2016[2]corpus

Systems	AUC on OIE2016
TextRunner	-
ReVerb	-
OLLIE	0.202
ClausIE	0.364
Stanford	0.079
OpenIE4	0.408
MinIE	-
RnnOIE	0.462
Seq2seq OIE	0.473
SpanOIE	0.491

2.5 Coreference Resolution

A document generally contains more than one sentence. Thus, each sentence can be contained within a given context. As mentioned in Section 2.3, allowing models capture a sense of context into the sentence (through models such as ELMo and BERT) brought state-of-the-art improvements in almost all NLP tasks. This need for context is relevant because of the many different forms and types of reference outside of a sentence that the model can not handle as input.

There are several types of references in the English language that make Coreference Resolution extremely complex. In Sukthanker et al. [106] 8 types of references are presented:

- **Zero Anaphora** is a reference that assumes a kind of listing. For example, in the sentence: *Pedro always carries 3 things (1):*, *his laptop (2)*, *his mobile phone (3)*, *his keys(4)*. It is possible to notice that (1) it refers to 3-anaphors;
- **One Anaphora** is a type of reference when it is used a similar expression such *that one*, where the demonstrative varies and the word *one* is maintained;
- **Demonstratives**, in general, assumes a comparison with something that appeared before. For example, when demonstratives are used like *than, that*;
- **Presuppositions** is a type of reference often used with the pronouns *anybody, everyone, amoung others*. This kind of reference can be very ambiguous. For example:*If someone (1) can do the task, it is him (2)*;
- **Discontinuous Sets** assume, similarly to Zero Anaphora, that the pronoun may refer to more than one antecedent. For example, when using the pronoun *they*, it can refer to *Pedro and Gisela*;
- **Pronominal Anaphora** is the most common reference used in everyday life, newspapers, and magazines. It can be divided into three major types: Indefinitive, Definitive and Adjectival Pronominals. For example: *Pedro is happy. He got a console.* the antecedent is *Pedro* and the anaphoric expression is *He*.

- **Cataphora** is also known as the opposite of anaphora. In this kind of reference, the anaphoric reference happens before: *He (1) was so generous to bring the dog, although Pedro (2) did not need to do it*.

- **Inferable or Bridging Anaphora** is one of the most ambiguous references because an understanding of context or previous knowledge is not necessary. For example, in the sentence *Pedro wanted to buy a pair of jeans (1), but he noticed a defective in the pockets. (2)*, (2) refers to (1), not directly, but by making background assumptions.

Since early on, the first algorithms, for coreference resolution, were created in 1978 [48] that used parse tree analysis to create rules capable of solving pronouns in the sentence. In 1974 [60], a system was developed a system using logic programming to analyze the sentences presented to discover noun phrase antecedents syntactically. Today, deep learning models are used for coreference resolution [122, 20] reaching state-of-the-art in the area.

Chapter 3

Related Work on Fact-Checking

Fact-checking, associated with Natural Language Processing, is a cluster of several NLP research areas allowing to create robust end-to-end systems. Thus, this chapter will start by focusing on the two main components that generally incorporate a pipeline of fact-checking (Information Retrieval and Question Answering), followed by the main challenges of executing such pipelines. The first component, *Information Retrieval*, is presented in Section 3.1 and studies on how to find and rank by relevance useful information in a given data store. Hence, it is an important component for *Document Retrieval* and *Evidence Retrieval* tasks. The second component is *Question Answering*, and is presented in Section 3.2. This section will explain how to analyse and extract information from a given question (i.e., claim) and corresponding answers (i.e., evidence). Finally, the dataset used in this dissertation will be explained, in SubSection 3.3.1, as well as current state-of-the-art approaches that explore it, in Section 3.4.

3.1 Information Retrieval

Information Retrieval (IR) is an area that studies mechanisms that can identify and rank information. This area is evaluated by, through an analysis of a given query, the ability to retrieve unstructured text that is relevant to the query. At the moment, a crucial step is the ability to map words in vectors with algebra models. Thus, this mapping enables the evaluation of the produced vectors with a similarity between a query and text through cosine-similarity [94]. Named Entity Normalization can resolve many problems in end-to-end information access tasks [56]. This means that the ability to disambiguate names like Bush as a person or Bush as a plant is a crucial task and can solve many issues found in Information Retrieval tasks. Also, focusing only on entities automatically neglects stop-words.

New approaches focus mainly on calculating the likelihood of a query being close to a document and thus presenting a document ranking. However, this ranking is extremely sensitive to a parameter called smoothing [128]. Smoothing adjusts the maximum likelihood estimator in order

to mitigate the data sparseness that provokes inaccuracy. This is very important when working with unknown sources.

There are many metrics to evaluate different IR mechanisms:

- **Precision** evaluates the capacity if the retrieved documents are correct, being important to assess if the top-ranked documents were retrieved correctly.

$$precision = \left| \frac{relevant\_documents \cap retrieved\_documents}{retrieved\_documents} \right|$$

- On the other hand, **Recall** assesses the ability if the relevant documents are retrieved. This metric is vital for assessing sensitivity, mainly when thresholds are used.

$$recall = \left| \frac{relevant\_documents \cap retrieved\_documents}{relevant\_documents} \right|$$

- **Fall-out (Noise)** is one of the most interesting metrics since evaluates the number of wrong documents that have been retrieved. This metric can be easily fooled if zero documents are retrieved. However, the higher this metric is, the more noise is being retrieved.

$$noise = \left| \frac{non-relevant\_documents \cap retrieved\_documents}{non-relevant\_documents} \right|$$

- **Discounted cumulative gain** is a metric which, if any document related to the query has not been removed, will penalise it in a more significant way. There is also a greater penalty the less likely the retrieved documents are to be relevant [117].

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

where p stands as a particular rank position.

3.2 Question Answering

Question Answering (QA) is a research area that focuses the attention on answering questions presented by humans in natural language. One of the first QA applications was developed in 1961 to be able to answer simple questions about baseball [42]. The LUNAR system is a remarkable step in the history of Question Answering by answering questions related to the geological analysis of rocks during Apollo expeditions [123]. These systems were able to answer 90% of the questions in its domain posed by people that did not know how the machine worked. A more recent mark is the IBM Watson being able to compete against humans [35]. The machine has a 64% win rate against humans in the *Jeopardy* contest.

QA architectures depend mainly on how the information is presented. From this, two major divisions are made: unstructured data and structured data. Plain text corresponds to unstructured

data, and the first approach to deal with unstructured type of data relies on Information Retrieval. Meanwhile, for structured data, databases, or even graphs, are the intermediate representations of this kind of data. Structuring data allows the use of more specific mechanisms, like Resource Description Framework (RDF), to extract the information and process it.

A base structure for QA models is followed, so a generic pipeline is defined with three major modules [85][50][21]:

1. **Question Analysis:** All questions are different. The input from the user is in Natural Language, and therefore its processing is required. Information extraction and classification need to be processed, analysed, and subsequently transformed in the best way to be used in the next phase. One of the first approaches, in 2003, performed stop-word removal, conversion of the inflected term to its canonical form, query term expansion, and syntactic query generation was implemented [19]. Phrase level dependency graph was adopted to determine question structure and the domain dataset was used to instantiate created patterns [124];

2. **Passage Retrieval:** Passage retrieval focuses primarily on an information search in which an engine is used that will lead to a retrieve of information significant to the question. This retrieval information is evaluated as passage candidates or sentences that are extracted from a knowledge base. From the question analysis module, this stage will formulate queries and looks up for potential answers using as a base the different information sources that can answer the question. Other unstructured information sources, where the information presented is dynamic, like the web and some online databases can also be used here. Being an IR mainly problem, some implementations can use knowledge graph information sources [98, 112];

3. **Answer Extraction:** Using the appropriate representation of the question and each candidate passage, candidate answers are extracted from the passages and ranked in terms of probable correctness. The main techniques used for Answer Extraction are the use of n-grams [29], patterns, named entities and syntactic structures. More recently, an Answer Extraction technique was introduced that uses a merging score strategy (MSS) based on hot terms [61].

Figure 3.1 shows a typical flow of a QA system's framework.

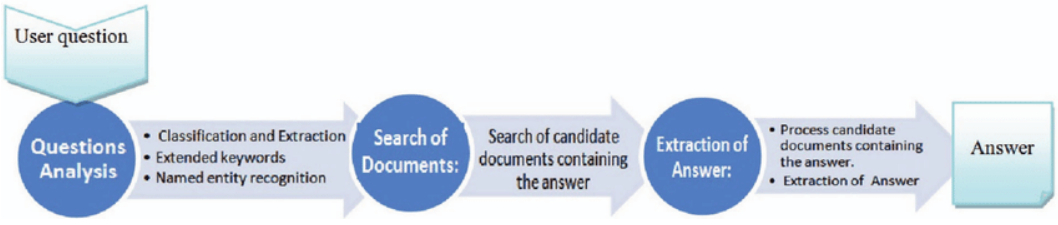

Figure 3.1: Framework of a QA system, adapted from (Bouziane, Bouchiha, Doumi and Malki, 2015) [1]

3.3 Challenges

Challenges usually have an overall goal. For those who propose the challenge, they need to have the ability to organise and give all the conditions to achieve the main objective; for participants, the ability to develop techniques that are innovative and make them better than others. There will never be a single solution, but the most important thing is the learning obtained from each developed system. Therefore, challenges for the fact-checking area are important in the sense that they foster progress in the development of computer systems to address this type of task.

Thus, in this section, we will reference FEVER, the dataset on which this dissertation will focus to validate its approaches. For this dataset, there were two challenges: FEVER 1.0 and FEVER 2.0. Then will be presented other noticeable datasets and their competitions.

3.3.1 FEVER Challenge

The main objective of the Fact Extraction and VERification (FEVER) Challenge is to create automatic systems that can verify pieces of information by extracting evidence from Wikipedia.

There have already been two iterations of the challenges:

- Fever 1.0 Shared Task (August 2018)[1] was the first challenge, where it focused only on creating automatic systems for the creation of fact-checking pipelines capable, above all, of finding the correct evidence [111];
- Fever 2.0 Shared Task (August 2019)[2], where adversarial examples were used to fool the systems developed in order to improve their robustness [109].

In order to validate the developed systems, a corpus was created with about 5.4 million Wikipedia articles and about 185 thousand claims. The systems would have to go through every claim and extract relevant evidence. In the end, it was necessary to classify the claim in three possible ways: *SUPPORTED*, *REFUTED* or *NOT ENOUGH INFORMATION*. If the claim were classified as *SUPPORTED* or *REFUTED*, in order to verify it, to support the given label is necessary to retrieve evidence.

The evaluation of the systems is done through a metric called FEVER Score. It is the accuracy whether the system can correctly classify the claim if, and only if, at least one of the proposed evidence is in the gold standard evidence, created by different annotators. The dataset is divided into two essential parts: Wikipedia Corpus and Claim Dataset.

The **wikipedia corpus** was extracted from a Wikipedia dump in June 2017. This dump generated 5 396 106 articles where only two pieces of information were kept: title and text. Regarding the text, due to their size, only the introductory text which summarises the entire article was maintained.

[1] http://fever.ai/2018/task.html
[2] http://fever.ai/2019/task.html

In order to relate the extracted evidence to the documents and, more specifically, to each sentence, the title was not pre-processed, and the text was divided into sentences and numbered in ascending order. The sentences, of each document, were pre-processed using the Standford CoreNLP [70]. The titles of articles, in Wikipedia, are concise, without containing any special characters. It is necessary to recognise that some of the titles already have the kind of entity they represent, since it gives a quick understanding on what the document is about. For example: 'Kamal Muara Stadium', 'Richard Haddock (1673-1751)', 'Turn It Up (Pixie Lott album)', 'Ann Richards (singer)' and 'Ann Richards (actress)'. Since there are docs with the same name but referring to different entities, to disambiguate them, Wikipedia adds some suffixes in a parenthesised way. This disambiguation is very important to take notice; for example, the title named Ann Richards can be referred to the actress or the singer. The titles contain, on average, 2.8 tokens.

The **claim dataset** [110] contains a total of 185 445 claims with an average size of 9.4 tokens. Each of the claims was manually built from 50 000 most popular articles. In order to have no previous knowledge or some kind of bias in the interpretation of the claims, the annotators who built the claims did not make evidence annotations. A second group of annotators tried to discover possible evidence, not by searching in the 50 000 pages, but the whole corpus, making possible the existence of more than one evidence in the dataset for each claim.

One of the strengths of this dataset is the freedom given to annotators to do cross-reference between documents. Thus, to be able to classify a claim, it may be necessary to relate more than one sentence or even in different documents. In Figure 3.2 it is possible to verify that, for the claim *The Rodney King riots took place in the most populous county in the USA*, two different documents must be found: *LosAngelesRiots* and *LosAngelesCounty*. Within these two documents, it is necessary to discover three relationships: that Los Angeles riots (the title of one of the documents) are *also known as the Rodney King riots*, that these riots *occurred in Los Angeles County* and that, in turn, Los Angeles county is *is the most populous county in the USA*.

The number of supported claims is 93 367, while the number of refuted claims is 43 107; the number of claims with Not Enough Information is 48 971. The dataset is divided into four categories: Training (an unbalanced set that is dedicated to train or fine-tune models), Development (a balanced set with 9 999 claims that are used for validation), Test (a balanced set with 9 999 claims without knowing the annotations) and Reserved (a balanced set with 19 998 claims used for the FEVER shared task challenge). It is very important to notice that claims that need more than one sentence to be verified (includes supported and refuted claims), represents 12% of the whole dataset. In Table 3.1 it is possible to see that the training set and the development (dev) set were carefully divided in order to preserve a specified rate of hard cases. So, it is possible to get a maximum of 87.5% accuracy in the training set for the supported category retrieving only one sentence, while by retrieving two sentences it is possible to get a maximum accuracy of 0.983. Claims that need more than 3 sentences represent 1% of the whole dataset. For example, the claim *Bonnie Hunt was in a movie.*, with the ID *88173* is annotated as requiring 43 sentences in order to classify it as *SUPPORTED*.

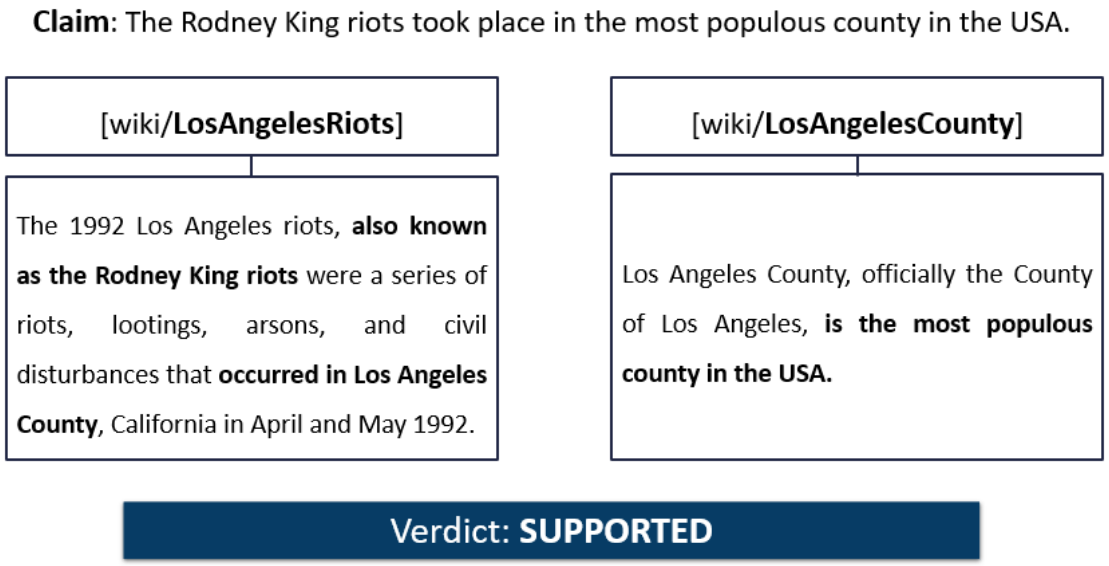

Figure 3.2: FEVER example where it is needed two documents to make a label classification correctly

3.3.2 Other Challenges

In this subsection, some prevalent challenges that can be related to the FEVER challenge are highlighted. This relationship is possible because, in general, in the Document Retrieval stage, the approaches and techniques are similar to those used in QA and Natural Language Inference (NLI) because of the way classification is done between two phrases – in FEVER, between a claim and evidence.

A vital benchmark to evaluate different NLP systems is the General Language Understanding Evaluation (GLUE) [116]. This platform is centred on nine English sentence understanding tasks, covering a broad range of domains, data quantities and difficulties, consisting of natural language understanding (NLU). Currently, there are 11 tasks evaluated. However, it is possible to evaluate the performance of the systems in 4 different categories: lexical semantics, predicate-argument structure, logic and Knowledge, and Common sense. Each of these categories is subdivided to evaluate them in extra detail.

The leaderboard[3] shows the performance of the different proposed systems in the 11 tasks and each category performed in a diagnostic dataset. The current state-of-the-art systems are mainly based on Bidirectional Encode Representations from Transformers (BERT) [28]. The highest ranked model is initialised in RoBERTa [65], a large model developed by Microsoft [53]. RoBERTa is leading most of the GLUE tasks with ensemble models based on it. Google recently submitted a new model named T5, based on transformers as well, reaching the highest score. However, there is no article published at the moment of writing this dissertation. Although it is not clear, this benchmark allows evaluating the strong and weak points of every model. It can also

[3] https://gluebenchmark.com/leaderboard

Table 3.1: Rate of sentences needed to prove a claim

		Number of Sentences from the same document			Number of Sentences from different documents	
		1	**2**	**3**	**2**	**3**
Training	**Supported**	0.875	0.170	0.025	0.152	0.024
	Refuted	0.901	0.142	0.019	0.132	0.018
Dev	**Supported**	0.899	0.145	0.017	0.128	0.017
	Refuted	0.921	0.116	0.015	0.102	0.014

allow for better real-life implementations, by having previous knowledge of what areas a model is good for.

Another relevant dataset is the well known Stanford Question Answering Dataset (SQuAD) [86]. The dataset is based on more than 100 000 questions posed by crowdworkers on a set of Wikipedia articles. Since the dataset is open, it presents itself as a challenge for research to overcome the human baseline performance of 86.8%. Similar to GLUE, leading research companies such as Google, Facebook, Microsoft, among others, have a high interest in reaching the highest performance on these tasks. Transformers such as RoBERTa and AlBERT [58] are reaching the best results merged with ensemble methods. Being a QA task, the models represent similar preprocessing methods in order to achieve the best document retrieval and reading comprehension performance (in a text, select the most meaningful words or sentences).

For the NLI area, the Stanford Natural Language Inference (SNLI)[9] was created. The purpose of this dataset is to understand how one sentence relates to another. SNLI contains about 570 000 English pairs of sentences that can be labelled as *entailment*, *contradiction* or *neutral*. A fundamental note, related to FEVER, is that the average number of tokens in the sentences inside the documents in this corpus is 14, compared to FEVER where this number is, on average, 31 tokens. Since it is possible to create relations between sentences, FEVER approaches took advantage and used models pre-trained on the SNLI dataset, mainly because the labels are similar. An improved dataset is the Multi Genre Natural Language Inference (MultiNLI) [119] that consists of 433 000 sentences, being a more diverse corpus that was designed to be incorporated as a task of the GLUE platform. MultiNLI, when compared to the SNLI dataset, provides a wider and more diversified range of text, alongside an auxiliary test set for cross-genre transfer evaluation

3.4 Automated Approaches on FEVER

In the FEVER 1.0 Shared Task, 24 teams participated, in which Thorne et al. [111] provide a summary of the main approaches that the teams made. It is interesting to note that almost all teams that exceeded the baseline score divided their end-to-end system into three stages: Document Retrieval, Evidence Extraction and Label Prediction. Since some teams wanted to remain

anonymous, the techniques used will be generally described, specifying whenever possible which team used which technique.

To implement a Document Retrieval stage, the main thing to consider is a reduction in the number of documents. With 5.4 million documents, it is vital to select only those that are relevant to the claim and reduce the amount of noise for the next stage. Similar to the first stage in the Question Answering task, a claim analysis needs to take place. The main mechanisms used in this stage are Name Entity Recognition, the extraction of Noun Phrases using Dependency Trees and the retrieval of all Capitalised Expressions. Some of the techniques, used in the FEVER 1.0 Shared Task, use TF-IDF to analyse the document's text and how important the text is for the claim to use unigrams and bigrams jointly. Other mechanisms developed make, in a first instance, the search solely by the title of the document, mainly for computational cost reasons.

In the stage of Evidence Extraction, the main objective is to select sentences that will serve as pieces of evidence for the claim. There is a maximum limit of 5 evidence sentences to be chosen. In general, the ESIM model [13], an enhanced LSTM for Natural Language Inference, was used with small adaptations, and there are even those who use an ELMo model. This model allowed the teams to focus on the problem as if it were binary. There was a team that used a cosine similarity over smooth inverse frequency weightings to evaluate the sentences as a similarity problem with the claim. It is essential to check the results, and thus UCL [78] team created a new aggregation stage in the pipeline where inconsistent evidence was discarded.

Finally, the stage of Label Prediction refers to the classification of the claim through the possible evidence extracted previously. The main methods used in the FEVER 1.0 Shared Task, were focused on Natural Language Inference methods, where it is possible to check models such as ESIM. However, the approaches were different, with some teams gathering the evidence sentences as one, others creating claim-sentence pairs or even creating a vote for each of the evidence sentence drawn. It is difficult to make a comparison between the different approaches because it depends directly on how the previous stage retrieves the evidence.

Table 3.2: Final results achieved in FEVER 1.0 Shared Task

Rank	Team Name	Evidence			Label Accuracy	FEVER Score
		Precision	Recall	F1		
1	UNC-NLP [78]	42.27	70.91	52.96	**68.21**	**64.21**
2	UCL Machine Reading Group [127]	22.16	82.84	34.97	67.62	62.52
3	Athene UKP TU Darmstadt [44]	23.61	**85.19**	36.97	65.46	61.58
4	Papelo [68]	**92.18**	50.02	**64.85**	61.08	57.36
...	...	...	...	...	...	...
12	Directed Acyclic Graph [88]	51.91	36.36	42.77	51.36	38.33
...	...	...	...	...	...	...
20	FEVER Baseline [110]	11.28	47.87	18.26	48.84	27.45

In Table 3.2, the principal scores obtained in FEVER 1.0 Shared Task by the different teams are presented. The **FEVER Baseline** [110] presented the initial benchmark that was available when the competition started. **Directed Acyclic Graph** [88] (DAG) was also included due to the

clarity on the steps and code available, being a better baseline than FEVER by using named entities to retrieve all the sentences from the documents, besides using the baseline TF-IDF technique for evidence extraction. The last step of DAG was built, using a Random Forest Classifier based on features extracted from the RTE model in order to label the claim.

The **FEVER baseline** system was developed by Thorne et al. [110]. The first steps are based on QA systems, followed by a feature-based model. Every stage of this system is described in detail:

- **Document Retrieval** This stage was inspired in the document retrieval component from Facebook's DrQA system [11]. This system uses a cosine similarity between binned unigram and bigram Term Frequency-Inverse Document Frequency (TF-IDF) vectors in order to reach the most similar documents to the claim.

- **Evidence Extraction** Similar to the previous stage, a cosine similarity between the claim and sentences was used, resulting in the similarity score. The top 5 sentences were chosen for the next stage.

- **Label Classification** In this stage was used a multi-layer perceptron [91] from the 2017 Fake News Challenge to transform the claim and the evidence as features. The RTE model used was a decomposable attention model [81] between the evidence and the claim.

The top-rated system was achieved by the team **UNC-NLP** [78]. The authors presented a Neural Semantic Neural Networks (NSMN) that is an adaptation of the ESIM model. It adds shortcut connections between the matching layer and the input, making the output stage simplified by the use of only one max-pooling and affine layer. A fundamental concept introduced was the relatedness score that evaluates the chance of two sentences being chained. Also, WordNet features were added in order to improve the score. This happened because the model gains a better awareness of semantic relationships by having previous ontological knowledge. The documents were also split in 10%, containing only the cases of disambiguation, called hard cases. This cases means that, in the title, appears in parenthesis information capable of differentiating the same entity. For example, *The Savages (band)* and *The Savages (film)*. Every stage of this system is described in detail:

- **Document Retrieval** In a first instance if the document's title appears the same as in the claim, these documents are retrieved. If no document is returned, the same process is done, not for the full title, but each token in the title. On average, 8 documents are retrieved just by doing this extraction. For the titles of the disambiguation documents, an NSMN model is used to choose the documents, which should be able to disambiguate the value of the claim entities.

- **Evidence Extraction** Each sentence contained in the withdrawn documents is evaluated using a binary model. The output indicates whether or not these sentences can potentially be used as evidence through a given probability. The 5 sentences with the highest probability are chosen for the next stage.

- **Label Classification** For this stage, the NSMN model is also used extracting features from the WordNet, besides the GloVe and ELMo embeddings. On top of that, the 5 sentences are used as input to the model at the same time, and having, as another set of features, a Relatedness score between the five sentences.

Athene UKP TU Darmstadt [45] had the best score regarding the Document Retrieval task, reaching an accuracy value of 93%. Entity linking with the claim and the document title was the main factor to achieve the high score.

- **Document Retrieval** The Athene UKP team, focused on extracting all the possible documents using only the title name of the documents. In a first instance, to analyse the claim and extract all types of entities, a heuristic was built capable of removing, through a constituency parser tree, all entities. After this phase, queries are created with each entity and made to a Wikipedia search engine called MediaWiki API [4]

- **Evidence Extraction** To accomplish this task, a model has been built that has as input the claim and the candidate sentence. The output consists of a confidence value for the given sentence to support the claim. This model consists of two ESIM models, where the two scores are concatenated using a modified hinge loss that ranks the possible sentence.

- **Label Classification** Selecting the 5 sentences with the highest confidence of the previous step, a new ESIM model is used which will have as input the 5 sentences, each one concatenated to the claim. The input is compressed into one vector using attention and pooling operations to finally reach a final max-pooling layer in order to have only one vector and a final multi-layer perceptron with three outputs, representing the confidences for each FEVER class.

Team Papelo [68] reached the highest score in FEVER regarding Evidence extraction. This was achieved mainly by matching capitalised expressions between the claim and the title of the documents. Important to notice that, for every sentence, the title name is added as a feature of the models. It was also the first team that used Transformer Networks [114] in the label prediction stage. With the output of the model, the system uses a set of rules to classify the claim instead of a machine learning process. Every stage of this system is described in detail:

- **Document Retrieval** To make the most of the baseline work that was done, Team Papelo, in order to discover the relevant documents, uses a vector space similarity TF-IDF. To complete this process, a NER system was applied to the claim. Thus, documents containing exactly those entities in the title or the first document that has the smallest Levenshtein distance [63] were also extracted. At the Data Understanding stage, it was found that most of the claims were about movies. So a rule was made: a document is also retrieved if the title "X (film)" exists, whenever "X" is retrieved.

[4] https://www.mediawiki.org/wiki/API:Main_page

- **Evidence Extraction** This stage is also based on the baseline system and selects the sentences using a similarity score based on the TF-IDF score between the sentences and the claim.
- **Label Classification** The ESIM model uses only one token in order to handle out-of-vocabulary words. So, with the transformer model, this is solved by allocating 10 000 more indices for out-of-vocabulary words. A hash is used to take from all words their corresponding index. Also, in the training process, the 3 classes are balanced out. They also use a set of rules, instead of a machine learning process, because the claim is labelled for each extracted evidence separately.

BERT for Evidence Retrieval and Claim Verification [102] did not participate in FEVER competitions, but currently has the second-highest FEVER Score with 69.7% and the current state of the art with a recall of 87.1% for Evidence Retrieval.

- **Document Retrieval** It uses the same Document Retrieval process as UKP Athene system as described above.
- **Evidence Extraction** Two approaches are mostly used: pointwise and pairwise. Pointwise uses both potential evidence and claim using cross-entropy loss as input. Pairwise uses two models, giving a confidence value to verify or not verify the claim, respectively. In the end, a Ranknet Loss or a Hinge Loss is used for classification. The result that reached state-of-the-art was Pairwise using Ranknet loss. The base model was BERT fine-tuned in this task.
- **Label Classification** It uses the same models as in the previous task (Evidence Extraction), but with a specific fine-tuning for this task and with a classification layer with 3 outputs.

Dreams [130] is the state-of-the-art in both Label Classification Accuracy and FEVER score, with 76.85% and 70.60%, respectively, at the time this dissertation was written. It uses RoBERTa and XLNet models which that are the current best transformer-based models for Evidence Extraction and Graph-Based Reasoning approach for the Label Classification task.

- **Document Retrieval** It uses the same Document Retrieval process as UNC-NLP.
- **Evidence Extraction** It uses a Pointwise approach to determine the confidence of the possible pieces of evidence. After retraining both RoBERTa and XLNet models, the first obtains a slightly better performance of 87.64% in recall.
- **Label Classification** The model has the 5 most relevant sentences as input. Based on semantic role labelling, it constructs a graph automatically. Two modules are used: one to calculate word embeddings contextualisation using graph-based distance based on XLNet and another representing graph components and their reasoning over the graph for learning representations.

3.5 DeFactoNLP

DeFactoNLP [88] is a system that was part of the first FEVER Shared Task. It achieved fifth place in F1 Evidence Score despite being 12th in the FEVER Score. This system participated in the fever challenge using the team name Directed Acyclic Graph. This system deserves a special emphasis since it is the target of study in this dissertation.

Many systems, including *BERT for Evidence Retrieval and Claim Verification*, have built rules to determine the final Label given to a claim. In the Label Classification phase, as it has as input a fixed number of x sentences, it gets, as output, x possible ratings for a given claim. DeFactoNLP, to deal with this output, extracts 12 features and builds a Random Forest model to get the final Label. Figure 3.3 shows the DeFactoNLP system architecture in detail.

- **Document Retrieval** DeFactoNLP determines the top-5 most similar documents based on a Term Frequency-Inverse Document Frequency (TF-IDF) comparison between document and claim. However, as stated at the beginning of this section, the title of the documents induces a perception of their content, as they refer to a given entity. A second pipeline is used, completely independent from the previous one. This pipeline employs a state-of-the-art Named Entity Recognition (NER) model to extract named entities mentioned in the claim. An additional document retrieval step is performed to obtain, for each entity extracted from the claim, a single document with the lowest *Levenshtein distance* [63] between its title and the entity.

- **Evidence Extraction** In this step, the system implements two separate steps that are a continuation from the two previous pipelines: (a) from the documents retrieved, the 5 sentences that are most similar to the claim are selected, each step based on TF-IDF scores; and (b) from the documents retrieved through NER, all sentences are included. In the end, all of the sentences are concatenated.

- **Label Classification** DeFactoNLP system achieves this Label Classification of a claim in a two-step process: (a) for every *claim-sentence* pair, it determines the label probabilities using a state-of-the-art model for RTE – the Decomposable Attention Model [81]; (b) to predict the claim label, it uses a Random Forest (RF) classifier based on the predictions made for each *claim-sentence* pair, and retrieves as *evidence* the 5 sentences of highest probability for the predicted label.

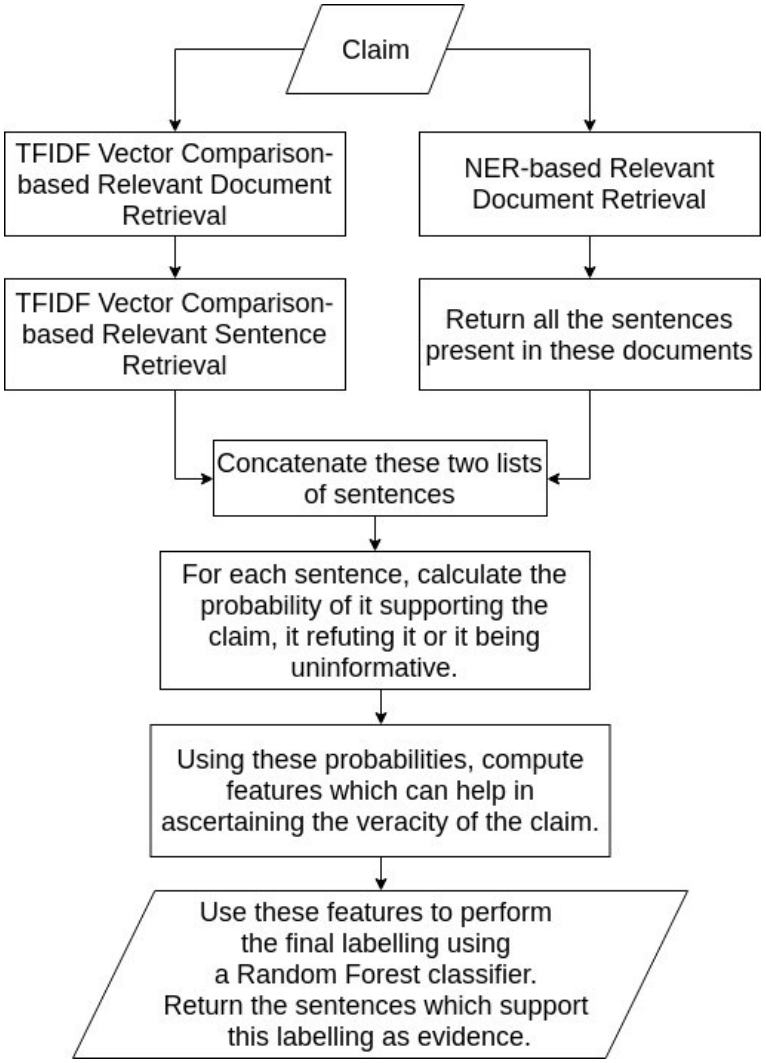

Figure 3.3: DeFactoNLP system architecture [88]

Chapter 4

Ablation Study on DeFactoNLP

In this chapter, the results obtained in the ablation study are presented for each of the three tasks described in Section 3.4. As explained in Chapter 1, this work will focus on the DeFactoNLP system as a baseline. Although not achieving the best results in the first FEVER Shared Task, the system reached the fifth best F1-Score in the competition and the architecture is clear and linear.

Since the main objective of this dissertation is the study of different Evidence Extraction Methods, it is essential to understand the current state of the system under study. Usually, an end-to-end fact-checking system is very sophisticated. Currently, the usual way to evaluate this type of system is through its final result: how accurately the system can classify the given claim, and whether it can provide relevant evidence that supports the classification label. The shortcoming of these systems is the lack of understanding where systems can be improved and/or are failing. With this level of fine-grained comprehension of the system, it is possible to point and make modifications that are capable of improving the final result. Also, an early-stage system error detection can help to mitigate possible error propagation that may occur.

The DeFactoNLP system is composed of three main task: (a) Document Retrieval, (b) Evidence Extraction, and (c) Claim Classification. Note that in tasks (a) and (b), two concurrent methods (TF-IDF and NER) are employed for each subtask, further increasing the complexity on the analysis of the system. In (c) a pipeline of methods is employed: a Recognizing Textual Entailment (RTE) Module followed by the Random Forest (RF) Classifier. Thus, it is peremptory to analyse with detail the current system. Because of this, an Ablation Study of the DeFactoNLP system was performed.

Document Retrieval, Evidence Retrieval and Label Classification are investigated in Sections 4.1, 4.2, and 4.3 respectively. In this exploratory work, changes are implemented in the pipeline, and the input data is also changed and controlled to evaluate the system. The FEVER score is also analyzed to draw different conclusions on how to improve and explore different techniques that might improve the system as a whole. The input relies on a sub-sample of the FEVER training set of 10 000 claims, obtaining a balanced dataset, because both the development (dev)

set and the test set are balanced. The experimental results are based on the dev set. As evaluation metrics for the Document and Evidence Retrieval components, Precision and Recall are used since they are the most relevant metrics in the Information Retrieval area [67] as was detailed in Section 3.1.

4.1 Document Retrieval

The first task in the DeFactoNLP system is Document Retrieval. As mentioned in Section 3.4, two techniques are used: TF-IDF and NER (Section 3.5). In order to perform an analysis of the data presented, it was necessary to make an initial understanding of the dataset. When different annotators analyse each claim, there might not exist a consensus on which sentences are considered as evidence. Consequently, there is also no consensus on which documents are considered relevant, even though there is always unanimity on the given label. Thus, each claim may be annotated with different evidence groups that can verify the claim. For example, in the claim:

- Edgar Wright is a person who directs.

Different groups of evidence can verify the claim:

1. **Document:** *Edgar Wright*
   Sentence: *Edgar Howard Wright (born 18 April 1974) is an English director, screenwriter and producer.*

2. **Document:** *Edgar Wright*
   Sentence: *Wright created and directed the comedy series Asylum in 1996, written with David Walliams.*

3. **Document:** *Edgar Wright*
   Sentence: *He is best known for his comedic Three Flavours Cornetto film trilogy consisting of Shaun of the Dead (2004),* ***Hot Fuzz (2007)****, and The World 's End (2013), made with recurrent collaborators Simon Pegg, Nira Park and Nick Frost.*
   Document: *Hot Fuzz*
   Sentence: *Hot Fuzz is a 2007 action comedy film directed by* ***Edgar Wright*** *who co-wrote the script with Simon Pegg.*

There are 3 different evidence groups that can label this claim as *SUPPORTED*. The first and second group of evidences contain only one sentence retrieved from a single document each; however, in the third group, two sentences are necessary to be linked which, in this case, are originated from two different documents.

To capture different properties in this ablation study, 3 different metrics were used to perform the analysis of this task:

- **Precision** is calculated by dividing the number of documents retrieved that are relevant, by the number of all the retrieved documents. Here an assumption is made where all the documents annotated are considered as relevant documents, not differentiating whether they

are in separate evidence groups. In DeFactoNLP, the Precision metric is relevant because of the existing discrepancy in the number of Documents extracted by the two techniques used: TF-IDF, in which there is a fixed number of 5 documents per claim, and NER, in which the number of documents per claim will vary according to the number of entities detected.

- **Recall** measures if the system can retrieve the relevant documents for a given claim. Mathematically, it is the division between the number of relevant documents retrieved by the number of all the relevant documents that have been annotated. The assumption made in the Precision calculation is reproduced for Recall. High recall values are very important in this task, otherwise relevant documents will not be considered in the following task of the pipeline system. By adding the two techniques together (TF-IDF and NER), the overlap will not be too significant to cover as many relevant documents as possible.

- **Accuracy** measures the number of correct results among the total number of claims. A correct result is considered when at least one group of evidence is contained in the retrieved documents. This metric allows understanding if, in the remainder of the system, the necessary documents to reach the correct label evaluating have been retrieved, in a certain way, the propensity of error propagation. The metric should have similar values to those of Recall, but only evaluates if at least one group of relevant documents is found or not.

With the metrics defined, the results of DeFactoNLP's document retrieval component are presented in Table 4.1 for both techniques, individually and in combination. Only the claims that are labelled as *SUPPORTED* and *REFUTED* are considered as input since the claims labelled as *NOT ENOUGH INFORMATION* do not have any evidence annotated. As expected, the Precision for NER is slightly higher than two times the Precision for TF-IDF. Since the titles of the documents usually represent entities, the NER technique obtained satisfying results for recall: 49%, similar to the 52% obtained by the TF-IDF technique. When combined, they reach a recall of 74%, showing that these two techniques capture different properties of the relevant documents, which end up being complementary. This Recall value means that 26% of the documents are still not retrieved, which demonstrates that improving the recall in this task is crucial for the system. When compared to the Recall obtained by UNC-NLP [78], DeFactoNLP presents a drop of 15% in recall. The accuracy values of DeFactoNLP were around 4% higher than the Recall. This shows that the system has a very high error percentage in this phase, creating another concern since this is the first task. Also, when compared to the UKP-Athene [45] system accuracy, the DeFactoNLP is 15% away from 93%. It was also noticed that, in about 10% of the claims, no entity was extracted (thus no document was retrieved).

Table 4.1: DeFactoNLP baseline - Document Retrieval

	TF-IDF	**NER**	**TF-IDF + NER**	**UNC-NLP**	**UKP-Athene**
Precision	0.166	0.353	0.180	-	-
Recall	0.524	0.491	0.741	0.890	-
Accuracy	0.557	0.522	0.783	-	0.936

4.2 Evidence Retrieval

After selecting all the documents, the most relevant sentences, that will be stated as possible evidence need to be retrieved. The input for this task corresponds to all sentences belonging to the documents retrieved in the Document Retrieval task. Therefore, this task drives by being able to select only the relevant sentences. In the Evidence Retrieval task, the DeFactoNLP system continues from the two previous pipelines (TF-IDF + NER from Table 3.3, so each pipeline has different groups of documents: from the TF-IDF documents selects the top-5 sentences and from the NER documents selects all the sentences. After these two pipelines, the sentences are concatenated as a group. To fully assess the performance of this task, using the same input and by freezing the Document Retrieval, Precision and Recall were used as metrics, and a change in the input was made. This change was made to evaluate the propagation of error in the pipeline architecture, by (a) using the input from the Document Retrieval task, determining the current performance of the task given the pipeline, and (b) the Evidence Retrieval performance independently from the system, by selecting the claims only when a correct group of documents of a evidence group were successfully extracted. This input was carefully reviewed to be balanced between the *SUPPORTED* and *REFUTED* claims.

Table 4.2: *DeFactoNLP* baseline - Evidence Retrieval

	TF-IDF	**NER**	**TF-IDF + NER**
Precision	0.104	0.091	0.078
Recall	0.429	0.498	0.699
Precision*	0.187	0.175	0.099
Recall*	0.770	0.952	0.892

* When at least one relevant group of documents is found

Table 4.2 details the results obtained by each of the steps employed for Evidence Retrieval, both in general and for the cases in which at least one group of documents has been found. Comparing the two pipelines separately, they both obtain a low precision. In fact, for documents retrieved using NER, all sentences are selected. Because of that, the Precision is lower than the TF-IDF approach, presenting an average of 12 sentences retrieved per claim. When combining all the sentences retrieved, the Recall is enhanced to 69.9% at the cost of reducing Precision to 7.8%, due to an increase in the number of selected sentences. A recall of 95% is possible to observe when at least one group of relevant documents is retrieved. This value is not 100% because not all groups of evidence are included.

4.3 Label Classification

The DeFactoNLP system has a two-step process to obtain a Label Classification for the claim (as explained in Section 3.4). Since this dissertation focuses on Evidence Extraction Methods, this task will be analysed as a whole. To perform this analysis, 5 evaluation metrics are used, as suggested in the FEVER 1.0 Shared Task [111]:

- **Strict Score**, also known as FEVER Score, was the main evaluation score used in the Shared Task. It considers the label accuracy if and only if at least one group of evidence was found. This score is the most important metric to evaluate an end-to-end system since it evaluates if the system is labelling correctly and if it has the relevant evidence to support it.
- The Label Classification **Accuracy** evaluates if the system gives the correct label to the claim. This metric can be crucial to understand if the system is correctly labelling the claims based on the evidence provided. This can be noticed by having very high accuracy and a low Strict Score.
- The **Precision** metric evaluates if the evidence selected by the system is relevant. In addition to the label that the Random Forest classifier gives to the claim, also only selects the sentences that were classified with the same label as the RTE model. Hence, Precision can give a perception on how the RTE model is performing.
- The **Recall** metric measures if the relevant evidence for the verifiable claims is found, where an instance is scored if and only if at least one evidence group is retrieved. The Recall may have values similar to the Strict Score. Since the Strict Score always needs correct evidence, a higher recall should mean a higher FEVER-Score.
- The **F1-Score** takes into consideration both Precision and Recall, providing a good perception on how this task handles the Evidence Selection.

Tables 4.3, 4.4 and 4.5 present the results for the Label Classification task, for the Dev dataset and also in the following settings, according to the observed results: those for which the Document Retrieval task finds at least one relevant group of documents ("Doc"), and those for which the Evidence Retrieval task finds at least one relevant group of sentences ("Sent"). These experiments were realized to evaluate the capability of the model by imposing that the previous pipeline worked correctly. It is important to understand how well the Label Classification models are.

Analysing the results in Table 4.3, the base FEVER Score of the system observed is 40%. One fact to take into account is the difference between Recall and FEVER Score. For "Dev", the difference is -1.14%, but for the "Doc" and "Sent" input, it improves by 4.7% and 5.2%, respectively. This evolution shows that the RF classifier has not learned the correct label based on the correct evidence. Precision can measure how well the RTE module can label the sentences. Therefore, even with an input with an average of 18 sentences as evidence per claim, the RTE module has a good performance over all the different settings.

Assessing the Recall in the Evidence Retrieval task (Table 4.2), there is a difference of 27% between the TF-IDF technique and the DeFactoNLP system baseline; however, its Precision is higher, revealing that TF-IDF retrieves less irrelevant evidences. Following this observation, an additional experiment for the *Label Classification* task was performed using only sentences retrieved employing the TF-IDF technique. The results are shown in Table 4.4. Against our expectations, the FEVER Score decreased from 44.1%, observed in Table 4.3, to 33%; however, Precision remained almost the same (50%), as compared to using the whole Dev dataset. Another crucial

Table 4.3: *DeFactoNLP* baseline - Label Classification

	Dev	**Doc**	**Sent**
Precision	0.515	0.559	0.570
Recall	0.386	0.490	0.533
FEVER Score	0.400	0.443	0.481
F1-Score	0.441	0.522	0.551
Accuracy	0.525	0.553	0.563

point is that for all the values, in the "Sent" setting, better scores are obtained in comparison with the baseline scores. This reveals that a higher precision in Evidence Retrieval is crucial for the Label Classification task. It also reveals that the RF classifier learns better with fewer sentences as input.

Table 4.4: *DeFactoNLP* baseline - Label Classification with TF-IDF as input

	Dev	**Doc**	**Sent**
Precision	0.499	0.578	0.600
Recall	0.278	0.499	0.572
FEVER Score	0.333	0.445	0.510
F1-Score	0.357	0.535	0.586
Accuracy	0.520	0.563	0.578

As observed in previous experiments, the RTE model presents considerable performance gains when the previous task in the pipeline provides less irrelevant sentences (a fixed number of 5 sentences). To correctly evaluate the impact of improving the precision of the models available in the Evidence Retrieval task, a RF classifier was trained with new input data. This training was performed in the same way as described by Aniketh et al. [88], using the training subsample as input, processed only by the TF-IDF pipeline. The RF classifier was trained with a maximum depth of 3 for the generated trees, a number of estimators of 50, and the criteria for classification type was "entropy".

Table 4.5: *DeFactoNLP* baseline - Label Classification with TF-IDF as input and Random Forest classifier retrained

	Dev	**Doc**	**Sent**
Precision	0.454	0.538	0.565
Recall	0.296	0.531	0.609
FEVER Score	0.322	0.468	0.537
F1-Score	0.358	0.534	0.586
Accuracy	0.525	0.600	0.612

Table 4.5 presents the results after this training. Although the Precision decreased, the RF classifier learned better when compared with the classifier trained with the baseline input. This is demonstrated by using "Sent" as input, which shows an increased Recall of 3%, a FEVER Score of 2.7% and an accuracy of 3.4% when compared to the non-retrained RF classifier. Once again,

this experiment demonstrates the importance of improving the Precision metric in the Evidence Extraction task.

4.4 Summary

In this chapter, an in-depth analysis of the DeFactoNLP system was performed, where each of its three tasks were dissected: Document Retrieval, Evidence Retrieval, and Label Classification.

In the Document Retrieval task, two concurrent pipelines were employed in the original DeFactoNLP system: TF-IDF and NER. Based on our analysis, in about 10% of the claims, no document was retrieved. An error analysis needs to be performed to find recurrent patterns in the data that lead to the low recall values observed in the system (e.g. to provide some insights for the claims in which the system does not retrieve any relevant document) Although the TF-IDF approach carries a content-based analysis of the documents, it only analyses the text through the frequency of certain words in the text. The use of embbedings should improve this content-based analysis to retrieve documents.

In the Evidence Retrieval task, the impact of the two pipelines from the Document Retrieval task was analyzed. The TF-IDF pipeline contains a fixed number of sentences, but the sentences that come from the NER technique do not present any kind of selection criteria (i.e. retrieving all sentences from the selected documents). Consequently, a variable number of sentences is retrieved. As verified during the *Label Classification* analysis, a fixed number of sentences and a reduced number of sentences as input from the previous tasks, improve the end-to-end system. Thus, a model capable of evaluating the sentences as relevant or not can impact the overall performance of the system.

Chapter 5

Improving Document Retrieval

The first task in the DeFactoNLP system is the *Document Retrieval* task. The aim of this task is to retrieve the documents that will be used in the following tasks of the pipeline to assess the claim and extract the evidence sentences that justify the final assessment. Given that *Document Retrieval* corresponds to the first task out of three in the pipeline, it is necessary to be able to find as many relevant documents as possible, hence, a high recall value, avoiding error propagation that can denigrate the performance in subsequent tasks. It may also affect the training of the models used, namely the Random Forest classifier, as seen in Section 4.3. In the ablation study performed in Section 4.1, a low recall of 74% was obtained, which compares with the 90% recall achieved in Nie et al. [78]. Therefore, 26% of the claims cannot be verified correctly.

In this chapter, the focus will be on the implementation of different techniques in order to improve the *Document Retrieval* process. In Section 5.1, Open Information Extraction (OIE) is presented as a promising approach to address disambiguation, which we explore as an alternative to Name Entity Recognition (NER). In Section 5.2, an algorithm to capture all the disambiguated documents and detecting them by using word embeddings is presented. Section 5.3 focuses on reducing the number of documents by analysing document content, title, and by exploring external data related to the pages. Finally, Section 5.4 frames *Document Retrieval* as a query problem to retrieve the relevant documents, using an external Application Programming Interface (API) named MediaWikiAPI used by the UKP-Athene team [45].

5.1 Exploring OIE

Taking into consideration the analysis provided in the Ablation Study presented in Section 4, the state of the *Document Retrieval* task of the DeFactoNLP system contains various imperfections. As mentioned in Section 4.1, one major flaw was detected: no entity is extracted in 10% of the claims. An error analysis was conducted to identify recurrent patterns of errors that are not being tackled by the proposed methods. To study what induced the errors, 25 verifiable claims were

randomly selected and manually analysed. For example, in the claim "Rabies does not affect the brain" the word 'Rabies' should be detected, since the document "*Rabies*"[1] is required to refute it; however, the NER approach is not able to detect this term. In Hanselowski et al. [45], a constituency parsing tree was used to analyse the claim by identifying certain words or group of words as terms. These terms were extracted based on hand-written rules. This dissertation explores whether Open Information Extraction (OIE), employed to structure the sentences as subject-predicate-object (S, P, O) triples can help to mitigate this problem. Going back to the provided claim, by using triple extraction techniques, the subject "*Rabies*" and the object "*brain*" are extracted. This structured data is potentially useful to retrieve the relevant evidence to validate or refute a given claim.

Additionally, another flaw was detected on the Error Analysis for the NER technique: documents with disambiguation terms were retrieved incorrectly. For example, in the claim "The principal photography of The Disaster Artist (film) started on December 8th, 2015." the film "*The Disaster Artist*"[2] is correctly extracted. Instead, the document that should be retrieved is "*The Disaster Artist (film)*"[3], because the former is a book. After investigating the claims, it was noticed that some of the claims disambiguate subjects and objects by including additional parenthesised information, in this case "*film*". To apply this information, a regular expression was used to search and retrieve the parenthesised information. The retrieved information is appended to the extracted subject or object it refers to before retrieving the closest document using the Levenshtein algorithm.

This step was missing in the baseline system, namely in the NER-based *Document Retrieval* approach. To perform OIE, the Stanford OpenIE[4] was used to extract triples. Since this method needed a complete triple to extract information, some claims did not have any triple extracted. So, to have a more complete triple extraction information, an implementation of ClausIE was used, using python + SpaCy [17]. To further extract some missing terms, for example, some movies or books terms, two main algorithms were taken into consideration: The Multi-Liaison Algorithm (Liaison) [54] and the Rusu Algorithm (Rusu) [93]. Multi-Liaison and Rusu algorithms have the same basis, although Multi-Liaison goes further in relations and can make more than one relation in one sentence, while Rusu only builds one relation and always identifies one even for intransitive verbs. So in this dissertation, these two algorithms were merged into one, as used in Azevedo et al. [3]. In a constituency parse tree, every sentence has 3 children: i) NP (Noun-Phrase); ii) VP(Verb-Phrase); iii) a single dot. In Figure 5.1, an example of the first three children is presented, using the AllenNLP demo[5] [37] tool that uses a model based on ELMo embeddings [55]. The analysed sentence is *An interesting pink panther has enjoyed a recent visit to Pedro's home.*. The NP of the sentence is *An interesting pink panther*, the VP is *has enjoyed a recent visit to Pedro's home*, and the dot is the end of the sentence symbol.

[1] https://en.wikipedia.org/wiki/Rabies
[2] https://en.wikipedia.org/wiki/The_Disaster_Artist
[3] https://en.wikipedia.org/wiki/The_Disaster_Artist_(film)
[4] https://nlp.stanford.edu/software/openie.html
[5] https://demo.allennlp.org/constituency-parsing

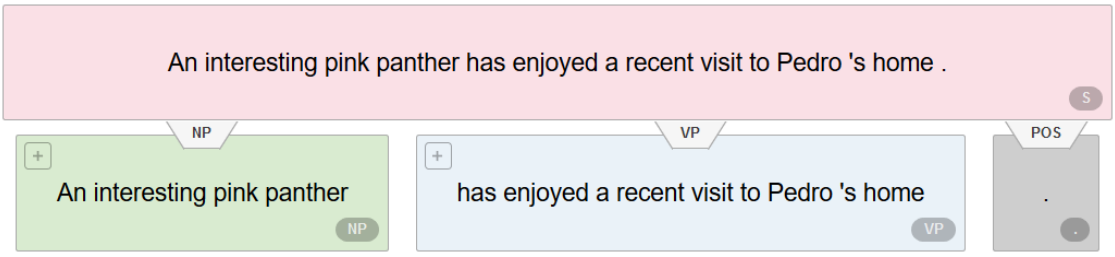

Figure 5.1: Visualization of the first three children in a Constituency Parsing Tree.

In order to identify the subjects of a sentence, a search for every NP subtree is performed. Each subject is found by performing a breadth-first search and selecting the first descendent of NP that is a noun. Nouns are found in the following subtrees [71]: NN (common noun, singular or mass), NNP (Proper noun, singular), NNPS (Proper noun, plural), NNS (common noun, plural), PRP (Personal pronoun) and CD (Cardinal number). In case of the existence of children starting with JJ (adjective) it means that this subject has an adjective. In Figure 5.2, by analysing the NP childs, it is possible to identify the subject *panther* with the NN tag, and the adjectives associated with the subject with the JJ tag *interesting* and *pink*.

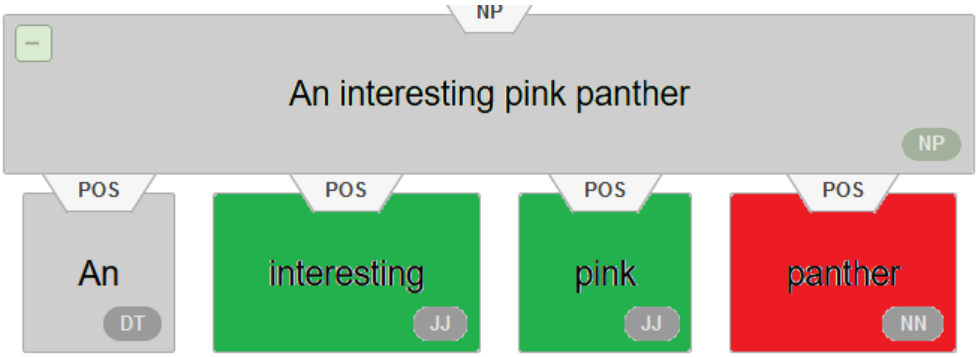

Figure 5.2: Constituency Parsing Tree visualization - subject extraction.

To identify the relation or to determine the predicate of the sentence, a search is performed in the VP subtree. The deepest verb descendent of the verb phrase gives the second element of the triple. Verbs are found in the following subtrees: VB (Verb, base form), VBD (Verb, past tense), VBG (Verb, gerund or present participle), VBN (Verb, past participle), VGP (Verb, non-3rd person singular present), VGZ (Verb, third-person singular present). In Figure 5.3, by expanding all the VP trees, the term *enjoyed* is selected as the verb identified as VBN. The term *has* was not selected since is not the deepest verb term.

In order to complete a triple, a search for the objects is done. These can be found in three different subtrees, all siblings of the VP subtree containing the predicate. The subtrees are PP (prepositional phrase), NP and ADJP (adjective phrase). In the two subtrees NP and PP, it is searched for the first noun, and if there are more NP siblings, the search for the first noun is done

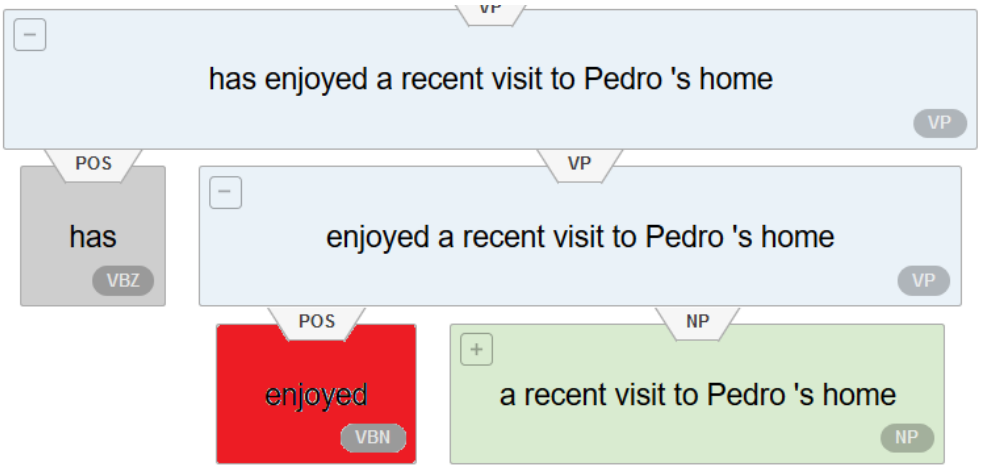

Figure 5.3: Constituency Parsing Tree visualization - relation extraction.

and added to the sentence, while in ADJP the first adjective is retrieved. Adjectives are found in the following subtrees: JJ, JJR (adjective, comparative), and JJS (adjective, superlative). In Figure 5.4, all the NP trees are expanded. The object found is the term *visit* with a NN tag with the adjective *recent*. After analysing the siblings, another two NP term that contain nouns. Therefore, the term *Pedro* identified as NNP and the term *home* identified as NN are also extracted as part of the object. Finally, the triple is completed: ($\langle panther \rangle$, $\langle enjoyed \rangle$, $\langle visit\ Pedro\ home \rangle$)

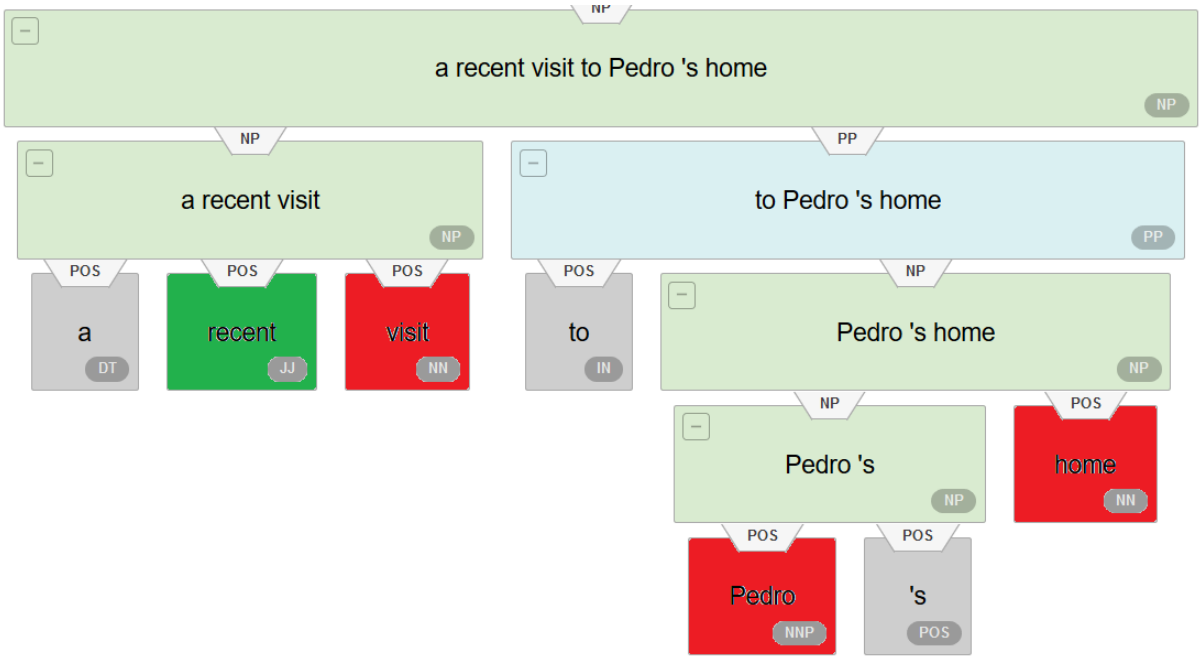

Figure 5.4: Constituency Parsing Tree visualization - object extraction.

Finally, Levenshtein distance [63] is used to extract the closest document for each (disambiguated) subject and object. Table 5.1 compares the results obtained by the baseline (TF-IDF +

NER), employing only the OIE and by merging the results from OIE with TF-IDF. Using only OIE achieved a recall of 66%, surpassing the 49% that was achieved only using Named Entity Recognition (Table 4.1). Concatenating OIE-retrieved documents with those from the TF-IDF approach, it is possible to observe a 6% increase in recall and a 6.6% increase in accuracy as compared to the baseline, with a precision decrease of 5.6%. These results are due to the higher number and also quality of documents retrieved by the OIE approach. It was also tested whether retrieving the second document with the smallest distance would increase the recall; however, it only contributed to a decrease in precision. This result was unexpected, but, one can attribute this to the fact that the Levenshtein distance is meant to find only the closest document and overcome some spelling mistakes. For example, in the claim "*Lockheed Martin is a car company*", one extracted term is "*car*". The two closest documents to the term are "*car*" and "*bar*". Hence, it is possible to conclude that the claims barely has no written mistakes; otherwise, the recall should increase.

Table 5.1: Results of Document Retrieval using OIE

	NER	**Baseline**	**OIE**	**OIE + TF-IDF**	
				1 doc	**2 doc**
Precision	**0.353**	0.180	0.187	0.124	0.079
Recall	0.491	0.741	0.662	0.801	**0.804**
Accuracy	0.522	0.783	0.711	0.849	**0.853**

5.2 Tackling Ambiguity using Word Embeddings

The use of Open Information Extraction (OIE) in the analysis of claims proved to be an improvement for the system. However, a recall of 80% remains lower than existing state-of-the-art approaches. Thus, an error analysis was performed to the changes introduced in Section 5.1, by randomly selecting 25 verifiable claims. In almost all cases, it was noticed that the disambiguation problem prevailed. For example, in the claim "*Savages was exclusively a German film*", the extracted document was "*Savages*"[6], instead of "*Savages_(2012_film)*"[7]. The main point to take out is that the way to handle disambiguated documents remains far from being addressed, and it is the main cause of errors. Hence, there is a need to propose a new approach.

Through the error analysis presented, addressing the disambiguation problem should overcome by analysing the content of the documents. In the DeFactoNLP system, the TF-IDF approach already performs a content-level analysis; however, the relationship between synonymous or similar words is not captured, since the approach is based on word frequency. Thus, a document to vector approach (doc2vec) was explored, which represents in vector space form the text presented in the document [62]. The doc2vec model was imported from the Gensim platform [89]. To train the model, using the same training subsample employed in Section 4.3, all documents that can verify the claims were selected. A stop word removal was perform in the document text. The model

[6]https://en.wikipedia.org/wiki/Savages
[7]https://en.wikipedia.org/wiki/Savages_(2012_film)

was designed with a dimension of the vectors of 300 and a minimum word count of 5, to remove words with few occurrences in the documents. Finally, the model was trained during 40 epochs with 90% of the dataset and validated in the remainder of it. To test the model, the documents were compared with the claim, and the five most similar claims were selected. The model achieved a recall of 23%. When analysing the generated vectors and the retrieved documents content, similar documents had lower distances than non similar, but distances between similar claims and a document were random and without any criteria. Therefore, it is possible to assume that the model was not able represent the claims in a vector space.

Since the previous results were not as desired, a new approach was explored. The Levenshtein distance is based on the calculation of the number of transformations needed from one input to another. There are 3 types of transformations: insertions, deletions or substitutions. The titles of documents, containing particular disambiguation, always appear at the end. By calculating the type and number of transformations performed, it is possible to extract all the documents that contain a given disambiguation. Through the terms extracted using OIE from the previous process, if it is only necessary to perform insertions at the end of each term, this document is selected as a possible candidate. There are also cases where, given a specific event or film, the title of the document is different from the extracted term. For example, in the claim "*The Armenian Genocide was the Ottoman government's systematic extermination of 1.5 million Armenians*" the extracted term may be *The Armenian Genocide*; however, the relevant document would only be "*Armenian Genocide*"[8]. Thus, all documents that require only insertions or deletions transformations at the beginning or end of each word of the extracted terms are eligible. Hence, it is avoided random documents by changing words or even adding words in the middle of the words.

The problem with the described algorithm is the high number of documents presented. To overcome this problem, an approach based on word embeddings was implemented to evaluate which title is most relevant to the claim. Thus, GloVe embeddings [82] were used using pre-trained word vectors on Wikipedia 2014 + Gigaword 5 creating 300 dimensional vectors (glove.6B.300d)[9]. However, the titles of the documents are not considered sentences, but rather a set of words representing a given information. Consequently, a special and different algorithm is needed to measure a relation between the title and the claim. To make a comparison between them, the following steps were performed:

1. Claim stop-words[10] removal
2. Tokenize the document title and claim
3. Map each title tokens and claim tokens to the corresponding word embeddings
4. For each title embedding $t \in T$:
    (a) calculate the cosine-similarity distance with every claim embedding

[8] https://en.wikipedia.org/wiki/Armenian_Genocide
[9] https://nlp.stanford.edu/projects/glove/
[10] stop-word list from Gensim [89]

(b) select the token embedding with the lowest distance

(c) add the distance to a distance list

(d) remove the token embedding

5. calculate the distances average from the distance list

Table 5.2: Results of Document Retrieval Tackling Ambiguations

	OIE + TF-IDF	**Top-5**	**Top-7**	**Top-10**	**No Threshold**
Precision	**0.1240**	0.0688	0.0573	0.0523	0.0215
Recall	0.8010	0.9217	0.9252	0.9301	**0.9396**
Accuracy	0.8490	0.9505	0.9572	0.9616	**0.9713**

Lastly, the top n documents containing the shortest distance are selected. If the title of the document is contained in the claim, it is automatically selected so that there are no zero distances. In Table 5.2, the results obtained by extracting the top-n documents per term from the claim are presented and the results obtained without any threshold ("No Threshold"). The results of the previous Section is also included (OIE + TF-IDF). Because the algorithm process is computationally expensive, it was only evaluated on 1000 claims out of the 9999 available in the Dev dataset. The results of the presented technique evolve positively in the metrics of Recall and Accuracy; however, the value of Precision worsens, performing a Recall-Precision trade-off. Thus, analyzing the approach with the highest Precision obtained is the "Top-5" column that is still a low value (as it presents an average of 15.6 documents retrieved per claim) when compared to the Precision of 12.4%, achieved in the previous Section. Despite the technique reaching state-of-the-art values with an accuracy of 95% and a Recall of 92.1%, the number of documents passed to the next task is very high.

5.3 Narrowing Document Retrieval

Although the method presented in Section 5.2 was impressive, reaching a maximum Recall of 97.1%, the average number of documents extracted was too high. This can lead to poor performance at the *Evidence Retrieval* stage, due to the high number of sentences to be evaluated. Combined with very low precision scores, this noise increases the probability of the models not performing with the expected performance. There are 3 ways to analyse if a given document is related to the claim: the title of the document, the content of the document, and using external data. The title of the document is what has been worked extensively in this chapter. The content of the document contains information on what the document refers to. By using external data, it is possible to retrieve information about the importance of a given document and its level of interest. In this way, 3 methods were created to evaluate the degree of relationship with the claim:

1. Comparison between document title and claim, using the word embeddings procedure presented in Section 5.2, with the difference that zero distances are taken into consideration.

2. Comparison between each sentence of the document and the claim using an available pre-trained languange model (bert-base-nli-mean-tokens[11]) to produce semantically meaningful sentence embeddings [90][12] calculating their distance, using cosine-similarity.

3. Number of web visits made to the document.

Approach 1 calculates a distance. This distance is normalised between 0 and 1 and transformed to represent a score, where 1 is the highest degree of importance of the document for the claim. Approach 2 transforms all the sentences of the document and the claim itself into sentence embeddings. Similar to the steps of point 1, the cosine-similarity distance between each sentence and the claim is calculated. The sentence with the lowest distance is considered the best performing since it means that is the similar to the claim. Approach 3 introduces external data information. This information was extracted through the dumps of wikimedia[13], processing the information from January 2019. The use of external information was suggested by Nie et al. [78]. Analysing one of the error cases identified, in the claim "Savages was exclusively a German film", the relevant document is "Savages_(2012_film)". In Figure 5.5 it is presented the name of the document followed by the number of views of the document. It is possible to acknowledge that the relevant document for the claim is the one with the highest number of views by far (18 233).

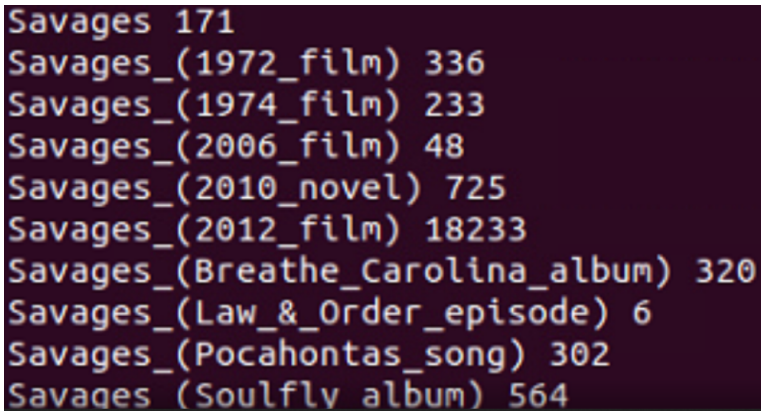

Figure 5.5: Page views of the disambiguated documents. In the left the document name. In the right the page views.

Table 5.3 presents the results of the above three approach used to narrow the retrieved documents. It is constrained the analysis on 1000 claims, since this process is computational expensive. It takes as input the documents extracted from the pipeline (top-5 for each entity), as described in Section 5.2. Glove refers to approach 1, Transformers to approach 2 and Page Counter to approach 3. For the Ensemble, k documents are selected from each process. The column with the Transformers values achieves a good Recall value. Although, using a transform-based model such as Bert, should be fine-tuned in the task [27]. Therefore, the Recall of 85,3% could be enhanced by

[11] pre-trained on SNLI [9] and MultiNLI [119] datasets
[12] https://github.com/UKPLab/sentence-transformers
[13] https://dumps.wikimedia.org/other/pagecounts-ez/merged/

Table 5.3: Results of Narrowing Document Retrieval

	Transformers Top-7	GloVe Top-7	Page counter Top-5	Ensemble 3+3+3*	Ensemble 5+3+2*
Precision	0.0968	0.1402	**0.1883**	0.1604	0.1394
Recall	0.8525	0.8959	0.8633	**0.9027**	0.9000
Accuracy	0.9013	0.9484	0.9103	**0.9538**	0.9520

* Ensemble order: Top-m from Sentence-Transformers + Top-n from GloVe + Top-p from Page Counter

fine-tuning the model on the FEVER dataset. Analysing the Page Counter column, it is possible to generalise that the relevant documents for the claim tend to be the ones with the highest number of visits. Retrieving the top 3 documents from each approach produced the best recall scores achieving 90% and an average of 7.4 documents per claim. The average should be nine since the ensemble model retrieves three documents from each technique, but the top k documents from the different approaches can retrieve the same document.

Table 5.4: Results of Narrowing Document Retrieval with TF-IDF

	GloVe + TF-IDF	PageCounter + TF-IDF	Ensemble (3+3+3) + TF-IDF (*)	Ensemble (5+2+3) + TF-IDF (*)
Precision	0.1157	**0.1498**	0.1237	0.1340
Recall	**0.9355**	0.9166	0.9230	0.9244
Accuracy	**0.9846**	0.9647	0.9728	0.9737

* Ensemble order: Top-m from Sentence-Transformers + Top-n from GloVe + Top-p from Page Counter

Table 5.4 follows the previous process, but the 5 documents from the TF-IDF pipeline were added. The best process consists of using GloVe + TF-IDF, reaching a 93.55% of recall and 98,46% of accuracy, with an average of 9.8 documents retrieved per claim. We attribute the improvement of the evaluation metrics to the complementary analysis of the TF-IDF, which retrieves relevant documents based on their content.

5.4 Exploring External Knowledge Sources

In the computer world, resources are an asset, and time is crucial. However, retrieving the documents and narrowing them as explained in Section 5.2 and Section 5.3, requires a prohibitive computational time, and, for the time of the completion of this dissertation, it was not possible to run the whole pipeline. Therefore, a new avenue was explored: with the OIE extracted terms (Section 5.1), use them as queries to make API requests to the MediaWiki API[14] to retrieve the documents.

Table 5.5 shows the results using the MediaWiki API by retrieving the top-n more relevant documents for every term extracted from the claim. Similar to the previous approaches, by increasing the variable n, the Recall also increases, but at a cost of a drop in the Precision. The

[14] https://www.mediawiki.org/wiki/API:Main_page

Table 5.5: Results of Mediawiki API for Document Retrieval

	SOTA	No Threshold	Top-1	Top-2	Top-3	Top-4
Precision		0.0247	**0.1451**	0.112	0.0703	0.0558
Recall	0.8923 [78]	**0.9338**	0.8842	0.901	0.9090	0.9154
Accuracy	0.9355 [45]	**0.9650**	0.9271	0.944	0.9475	0.9534

model that achieved state-of-the-art results was developed by Nie et al. [78] having a Recall of 89.23%. This model retrieved an average of 10 documents per claim. Using our approach, by retrieving the "Top-2" most relevant documents, as seen in the Table 5.5, the Recall obtained is 90% outperforming the state-of-the-art system by 0.9%, since the average number of documents retrieved is also 10.

5.5 Summary

In this chapter, we explored how to improve specifically the Recall in the *Document Retrieval* task. To replace the NER technique already implemented in DeFactoNLP system, Open Information Extraction methods were explored. The Recall increased from 49.1% to 66.2%. Aggregating the retrieved documents with the ones retrieved with the TF-IDF technique, already included in the DeFactoNLP system, a recall of 80.1% was achieved. We noticed that some of ambiguities observed while retrieving relevant documents for a given claim could be solved by exploring the disambiguation of terms presented in the Wikipedia titles. We propose an algorithm (Section 5.2) to address these cases. Although the Recall achieved was 94%, the precision is very low, introducing excessive noise in the pipeline. Thus, approaches capable of selecting only the most relevant documents were explored, namely: using word embeddings (to compare titles), using transforme-based models (to analyse the document content) and using the number of web visits to the document as extra feature. Despite reaching a Recall of 90.3%, the computational cost was too high to run for all the dataset claims. However, the use of external data, such as the number of document visits, proved to be important data to determine whether a given document is relevant for the claim. Thus, an External Knowledge Source to retrieve documents using generated queries from the OIE analysis was explored. The obtained Recall was 90.1% achieving state-of-the-art results.

Chapter 6

Improving Evidence Retrieval

The *Evidence Retrieval* task focuses on retrieving possible evidence from the previously retrieved documents. The DeFactoNLP system has two pipelines analyzed in Section 4.2: TF-IDF, which retrieves 5 sentences, and NER, which makes no selection, so it extracts all sentences from the retrieved documents. The analysis carried out in Chapter 4 showed that the Label Classification models rely on the precision of *Evidence Retrieval*. Recent transformer-based models demonstrated good performance on tasks similar to Evidence Retrieval such for Question Answering (Section 3.2); therefore we hypothesize that exploring these methods might improve the recall in this task. It is also evaluated if the new methods, described in the previous section regarding Document Retrieval, have a positive impact on the system. The Random Forest classifier in the *Label Classification* task is always retrained for the new pipeline. This is a consistent change from the DeFactoNLP system that improves the end-to-end system as seen in the Ablation Study (Section 4).

In Section 6.1 the creation of a selection model based on Open Information Extraction is presented. In Section 6.2 and Section 6.3 a transformer-based model – BERT – is explored. It is also explored the fine-tuning of the model, with the creation of a new dataset. In Section 6.4 a new Wikipedia dump is created with the resolution of coreference chains, exploring state-of-the-art coreference resolution systems. Every section will include the results of the Label Classification to acknowledge the impact of the explored methods.

6.1 OIE model

In the ablation study, it was noticed that *Evidence Retrieval* precision might be a relevant performance factor for *Label Classification*. DeFactoNLP does not apply any filtering function to select sentence candidates. Adding a filtering function has the potential to increase the precision of the model [31]. The main goal is to create a model to increase *recall* of *Document Retrieval* and precision in the *Evidence Retrieval* phase. The model aims to decide whether a sentence $p \in \mathscr{P}$

is likely to be relevant to (i.e., supports or refutes) a given claim $\mathscr{C}$. Therefore, this binary model outputs if p should be considered or discarded. To leverage OIE information, a Random Forest model (RF) is trained using several features extracted based on Open Information Extraction triple (*subject*, *predicate*, *object*) extraction methods [31].

To generate triples based on the claims, ClausIE [26] was used. For training purposes, a dataset was created based on the original training subsample. The features vector was constructed by extracting the triples $\mathscr{T}_c$ per claim with four main types of features:

- Negation
- Similarity
- Lexical
- Occurrence

The summary of these features is shown in Table 6.1. The negation features are based on the distance of $\mathscr{T}_c \in p$ where is evaluated how similar is every $\mathscr{T}_c$ to contain a negation. The similarity type of features, calculates Smith-Waterman [100], Jaccard [79] and Cosine distances between c and p. This distances are proposed by Esteves et al. [31] in order to extract features to evaluate similarity between two sentences. The Lexical counts and analyses the presence of a certain type of characters. The occurrence is a set of binary features that evaluates whether or not each triple component is $\in p$. In the feature extraction method, the similarity and occurrence for each triple $t = \langle s, p, o \rangle$ was computed as well as its *relaxed* version, which comprehends the objects without special characters. Similarly to Gerber et al. [39], a total of 37 triple-based similarity features were implemented. The features were also implemented both for the exact match of each triple *object* as well as its relaxed version, where a θ allows having a match to similar strings (e.g., "USA" $\simeq$ "U.S.A.").

Table 6.1: Triple-based Extraction Features

Feature	**Description**
Negation	Negation distances s, p, o
Similarity	Jaccard, Cosine, Smith Waterman
Lexical	Total of tokens, Alphanumeric Digits, Uppercase, Special char Stopwords
Occurrence	($\langle s \rangle$, $\langle s, o \rangle$, $\langle s, p, o \rangle$) in p

The model was trained on 6 189 sentences in a balanced set, extracted from the subsample of the training set. Validated on 1 857 sentences, this model achieved an F1-Score of 84.7%.

In Table 6.2, an optimal result is possible to observe: the removal of 33% non-relevant sentences by implementing the Triple-Based model (TB) when compared to the sentences retrieved by the baseline of the DeFactoNLP system, although it was accompanied by a drop of $7,6\%$ points

in Recall. In column "TF-IDF & OIE+ TB", the scores presented obtained using OIE & TF-IDF in the *Document Retrieval* task (as detailed in Section 5.1) together with TB. It obtained a higher precision and a slightly lower recall, better scores in terms of precision, an increased ratio of 30%, and a drop of 2% in the recall score when compared to the "Baseline" approach, but 6% higher when compared to the "Baseline + TB". The last two columns, " MediaWiki + TB" and " MediaWiki + Top-10 TB" represent the results of ER using the documents obtained using the MediaWiki API as explained in Section 5.4. Given a claim, the number of sentences retrieved by the Triple-Based model is not constant. Therefore, the top-10 sentences are also presented that were retrieved based on the model confidence. The previous approaches did not retrieve a fixed number of sentences, since the model was trained as binary model. As expected, the MediaWiki + TB increases 6% recall to 75.9% when compared to the Baseline, while increasing the precision as well. Unfortunately, using a constant value of 10 sentences retrieved only achieves a recall of 70.5%.

Table 6.2: Results for Evidence Retrieval based on Triple-Based model (TB)

	Baseline	**Baseline + TB**	**TF-IDF & OIE + TB**	**MediaWiki + TB**	**MediaWiki + Top-10 TB**
Precision	0.078	**0.117**	0.101	0.091	0.090
Recall	0.699	0.623	0.681	**0.759**	0.705
Precision*	0.099	0.150	0.119	0.094	0.094
Recall*	0.892	0.795	0.802	0.790	0.734

* When at least one group of documents is found

To analyse the impact of the Triple-Based model in *Evidence Retrieval* task proposed in the overall FEVER score, five pipelines were evaluated:

a) the *DeFactoNLP* baseline calculated in the Ablation study, which does not include any Evidence Retrieval method ("Baseline");

b) the addition of the developed Triple-Based model to the baseline ("Baseline + TB");

c) changing the *Document Retrieval* using TFIDF + OIE and *Evidence Retrieval* using the Triple-Based model ("TFIDF & OIE + TB").

d) using the MediaWiki API for the *Document Retrieval* and *Evidence Retrieval* using the Triple-Based model ("MediaWiki + TB").

e) using the MediaWiki API for the *Document Retrieval* and *Evidence Retrieval* by retrieving the Top-10 sentences from Triple-Based model ("MediaWiki + Top-10 TB").

The results for each pipeline are presented in Table 6.3.

Employing the proposed Triple-Based method improved precision while just slightly reducing recall for the *Evidence Retrieval* task. As a consequence, the precision in *Label Classification*, when comparing the results obtained in pipelines (a) and (b), increased 4% while maintaining the

Table 6.3: Results on the Label Classification based on Triple-Based model (TB)

	Baseline	**Baseline + TB**	**TFIDF & OIE + TB**	**MediaWiki + TB**	**MediaWiki + Top-10 TB**
Precision	0.515	0.555	**0.575**	0.545	0.524
Recall	**0.386**	0.375	0.383	0.334	0.374
FEVER	0.400	0.398	0.392	0.391	**0.401**
F1-Score	0.441	0.447	**0.456**	0.414	0.436
Accuracy	**0.525**	0.523	0.507	0.511	0.521

same FEVER score. In pipeline (c), recall in the *Document Retrieval* task is the highest comparing with (a) and (b) and, on the *Evidence Retrieval* task, a similar recall to the pipeline (a) and almost equal precision to the pipeline (b), combining the best of both pipelines. These improvements are reflected in Label Classification with an increase of 6% in precision and 1.5% in F1-Score. In pipeline (d), recall in the *Document Retrieval* task is the highest amongst the proposed pipelines and also the highest recall in the *Evidence Retrieval* task, but it presents the worst FEVER-Score. This can be related to a lower precision in *Evidence Retrieval* task. The number of sentences returned by the Triple-Based model is not controlled, returning a variable number. To assess whether this might be a problem for the *Label Classification* models, a fixed number of sentences to be returned was set to 10. The results are reproduced in the pipeline (e). Although the recall is not as high in the *Evidence Retrieval* task (less 5.5 %), the FEVER-score increased, as well as the recall in the *Label Classification* task. These results show that the model learns best if there is a fixed number of sentences as input, which seems to impact the performance of the RF model.

All of these results reveal the impact that a good recall in the *Document Retrieval* task and an even better precision in the *Evidence Retrieval* task have in the overall performance of the system: a wider selection of documents and a narrower selection of sentences can improve the *Label Classification* task.

6.2 Pairwise Evidence Retrieval

The most recent studies show that the use of transformers in the most varied tasks of NLP reaches state-of-the-art results [27, 65, 58, 125]; however, produces out-of-the-box rather inadequate sentence embeddings [32]. In Reimers and Gurevych [90] present an architecture to produce semantically meaningful sentence embeddings. These embeddings can be used efficiently for semantic similarity tasks; therefore, they can semantically assess the sentences most similar to a given claim. In Figure 6.1 it is possible to see schematically how the models provided can produce embeddings for the claim and the sentence separately. Afterwards a cosine-similarity is calculated to provide a score between 0 and 1 to evaluate how similar the claim and the sentence are.

The first experiment, to implement in the DeFactoNLP pipeline a transformer-based model, used an available BERT base pre-trained model ("bert-base-nli-mean-tokens") on SNLI [9] and MultiNLI [119] datasets. The BERT base model was chosen since it requires the least computer

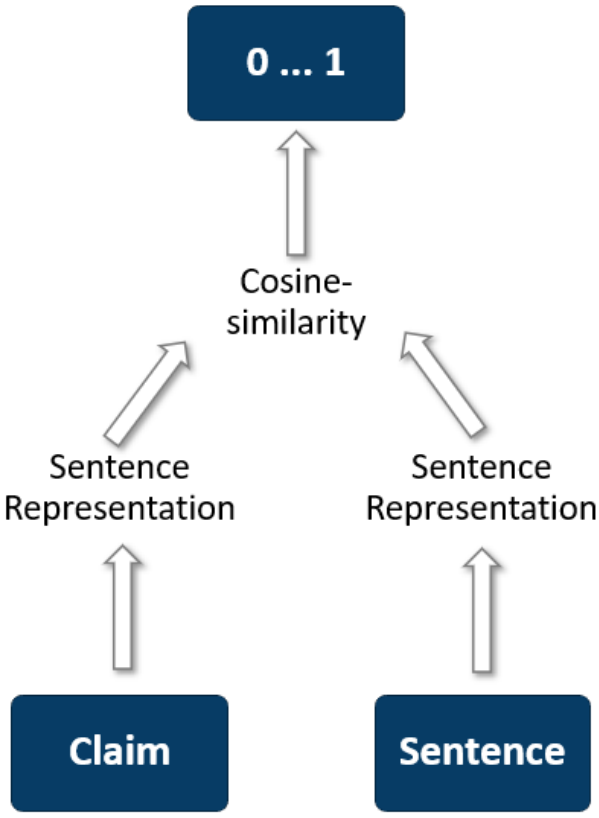

Figure 6.1: Pairwise Architecture to produce similarity score between a claim and a sentence.

resources to run amongst other transformer-based models. Therefore, the BERT base model is used as our base model to create comparisons between experiences. For the FEVER dataset, it was fine-tuned on 14 265 relevant and 14 164 non-relevant claim-sentence pairs for 3 epochs. This claim-sentence pairs were extracted using the our train subsample, from the relevant sentence. Although state-of-the-art method, decreased our recall from 69.9% to 49%. When using another transformer pre-trained model on the same datasets ("roberta-base-nli-mean-tokens"), it was achieved a recall of 63.4% and a FEVER-Score of 41.1%. This result is due to the fact that is a bigger model and a more complex one, allowing to capture the relations between sentences better.

The fine-tuning used was insufficient, leading to a need to create a larger and more complete dataset consisting of a claim, a sentence and a given label. This label is a binary value where 1 represents that the claim is related with the sentence, and 0 represents that the claim is not related with the sentence. For the 0 label, the sentences are selected by retrieving one random sentence from the documents that contains the sentence that can verify the claim. Hence, based on the training samples of the FEVER dataset, the following three datasets were created:

a) Contains 203 932 related pairs and 144 260 non-related pairs. This dataset is quite unbalanced, pending to the related pairs.

b) Contains 203 932 related pairs and 519 793 non-related pairs. This dataset is also unbalanced, pending to the non-related pairs.

c) Same dataset as b) but adding the name of document where the sentence was retrieved. This was done to imply some document context to the sentence.

Based on the same pre-trained model ("bert-base-nli-mean-tokens"), it was fine-tuned using each created dataset. The model was trained during one epoch as suggested by the article authors in Reimers and Gurevych [90]. Since the fine-tuning was on a different dataset, warm-up steps were performed with 10% of the training dataset. It was trained with a batch size of 16 with a maximum of 500 input tokens. The learning rate was $2e^{-5}$, with a weight decay of 0.01, with warm-up over the first 10 000 steps, and a linear decay afterwards. The Adam [57] optimiser is used with a linear learning rate decay. The learning rate used was based on the default recommended settings by Devlin et al. [27], and the warm-up steps was based on the recommendation by Li and Hoiem [64] for the model learn and not forget when trained on another dataset.

Table 6.4: Results on the Evidence Retrieval using Pairwise approach

	No Tuning	**a) No Title**	**b) w/ Title**	**c) Augmented**
Precision	0.065	0.077	0.080	**0.086**
Recall	0.510	0.579	0.610	**0.657**
Precision*	0.068	0.081	0.084	0.090
Recall*	0.531	0.603	0.635	0.684

* When at least one group of documents is found

Applying the fine-tuned models in the dev dataset, in Table 6.4 the results on the *Evidence Retrieval* task are presented, for each of the fine-tuning models using the MediaWiki technique for *Document Retrieval*. Every model retrieves exactly the top-10 sentences. The results without the pretrained model ("No Tuning") achieved a recall of 51%. The fine-tuned model in dataset a) improved the model increasing the recall by 6.9%. Introducing the document title, contributed to increase even further the recall by 3.1%. Lastly, fine-tuning on the dataset unbalanced towards the non-related pairs of sentences improved the recall by 4.7%.

Table 6.5: Results of Label Classification using the Pairwise Model

	Baseline	**Triple-based Model**	**Pairwise Model**
Precision	0.515	**0.524**	0.508
Recall	0.386	0.374	**0.401**
FEVER Score	0.400	0.401	**0.414**
F1-Score	0.441	0.436	**0.448**
Accuracy:	0.525	0.521	**0.538**

Incorporating the pairwise model into the system, to perform an analysis in the *Label Classification* task, achieves the results presented in the Table 6.5. All the metrics improved and almost at the same rate, with exception on the evaluation metric precision, which decreased. The most

important result is the FEVER Score, which increased 1.4% from the baseline. Individually, the recall also increased 1.5% and the accuracy 1.3%. These values can be related to two reasons: (i) the use of a fixed number of sentences retrieved (in this case, 10) enables a better retrain of the Random Forest classifier in the *Label Classification* task, and (ii) the increase of recall in the *Evidence Retrieval* task.

6.3 Pointwise Evidence Retrieval

A Pointwise approach has jointly the sentence and the claim as input. With this input will allow the model to classify the relevance of the sentence to the claim. While in the Pairwise approach, the generation of embeddings was performed in parallel, the pointwise approach takes into account all the input to achieve its result. In Figure 6.2 the created architecture is presented. The goal is to use the pre-trained BERT model and fine-tune using the proposed architecture. The BERT base model [27] was used (similar to in Section 6.2). This model contains 12 hidden layers, and we select the [CLS] token from the last hidden layer to feed into a Linear Layer, followed by a Softmax layer which is used to classify the sentence-pair as evidence or non-evidence. A cross-entropy loss is used with a learning rate of $2e^{-5}$ during 4 epochs, without decaying and no warmup steps. There were no warmup steps since the model is not pre-trained in any specific task. The dataset is split in two sets: 90% as a training set, and the remaining 10% as a validation set. An early stopping mechanism is applied everytime the validation loss increases when compared to the previous epoch.

To perform the fine-tuning, it was used only the dataset c) described in Section 6.2 that uses the name of the document as input and is an unbalanced dataset with 2.5 times more non-related pairs than related ones. This dataset was the only one used in this experience since it is the dataset that produced the best results, by far, in the previous experiences. The results of employing the new fine-tuned model into the *Evidence Retrieval* task, are presented in Table 6.6. All the results correspond to the setting where the 10 most similar sentences are retrieved. The pipeline uses the MediaWiki approach for the *Document Retrieval* task. For an easier comparison, the results of the previous models are also presented in the table. The results outperform the Triple-Based model, increasing recall by 0.9% to 76.8%, and precision to a ratio of 1.2. This last increase is the most significant improvement.

Table 6.6: Results on the Evidence Retrieval using the Pointwise Model.

	Baseline	**Triple-based Model**	**Pairwise Model**	**Pointwise Model**
Precision	0.078	0.091	0.086	**0.102**
Recall	0.699	0.759	0.657	**0.768**
Precision*	0.099	0.094	0.090	0.106
Recall*	0.892	0.790	0.684	0.798

* When at least one group of documents is found

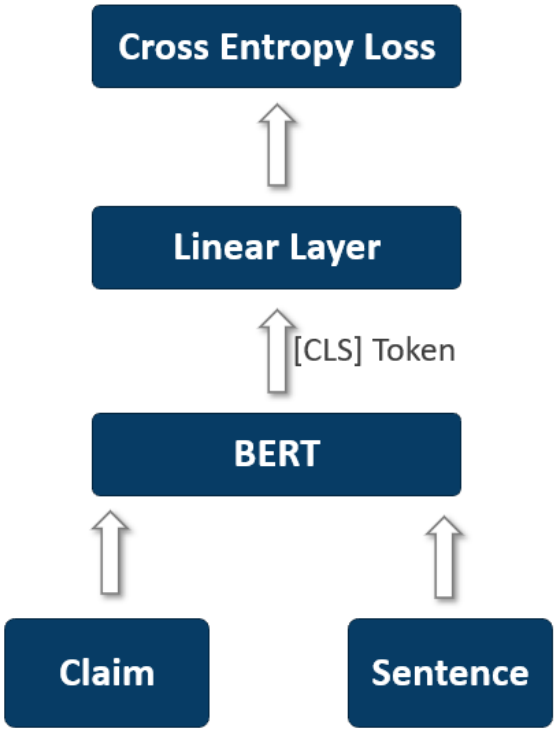

Figure 6.2: Pointwise Architecture to produce similarity score between a claim and a sentence.

Table 6.7 presents a comparison of the results of *Label Classification* including the new pointwise approach. With the pointwise model, it achieves the best results so far increasing the FEVER-score to 43.9%, an increase of 2.5% when comparing to the pairwise approach. An interesting result presented is the reduction of precision, something that had already been highlighted when analyzing the Pairwise model, while all the other metrics display improvements. Some reasons for this precision decrease may be related to the fact that the retrieved sentences are difficult to distinguish due to the complexity of transformer models. As a consequence, the correct semantic relations might not be correctly extracted, leading to a failed classification of the RTE model system in the *Label Classification* task. This does not happen in the Triple-Based model because, being a simpler model, the RTE model is able to capture the semantic relations capable of classifying the sentences in a more useful way.

Table 6.7: Results of Label Classification using the Pointwise Model.

	Baseline	**Triple-based Model**	**Pairwise Model**	**Pointwise Model**
Precision	0.515	**0.524**	0.508	0.476
Recall	0.386	0.374	0.401	**0.439**
FEVER Score	0.400	0.401	0.414	**0.439**
F1-Score	0.441	0.436	0.448	**0.457**
Accuracy	0.525	0.521	0.538	**0.552**

6.4 The Impact of Coreference Resolution

After the results of all models were presented, the Triple-Based model did not perform well in the *Label Classification* task and in the *Evidence Retrieval* yield lower scores when compared to the Pointwise model. Additionally, transformer-based models increased their results when given a certain context, in this case, the title of the documents (Section 6.2). To have a better understanding of the common mistakes made by the Triple-Based model, 25 cases where the system failed were chosen. This analysis allow us to understand, in a general way, recurrent patterns of erroneous predictions:

- The absence of a given entity. For example, given the claim "***Henry II of France*** *was not involved with the cause of a war about religion*", the correct evidence to support the claim is "***He*** *was succeeded in turn by three of his sons , whose ineffective reigns helped to spark the French Wars of Religion between Protestants and Catholics*". This is the case that impacts the TB model the most, since the entity is not part of the sentence being referenced anaphorically.

- Different mentions to the same Named Entity. For example, given the claim "***Edgar Wright*** *is a person who directs*", the sentence that supports it starts with *Edgar* ***Howard*** *Wright*. The word *Howard* makes the Triple-Based model fail in this case. Another example is in the claim "***Joe Rogan*** *had an acting career*", for which the supporting sentence only represents the entity as *Rogan*: *"After relocating to Los Angeles in 1994 , "****Rogan*** *signed an exclusive developmental deal with Disney , appeared as an actor on the television sitcoms Hardball and NewsRadio, and worked in local comedy clubs*".

- The choice of sentences that should be considered as evidence. Apparently, of the 25 sentences analyzed manually, 7 of them were missing annotations. For this reason, as seen in the *Document Retrieval* chapter (Chapter 5), the use of external data is so important, because the sentences noted belong to documents with a higher volume of visits. The claim "*Ernest Medina was uninvolved in a Vietnam War mass killing.*" is refuted by the following sentence: "*The "Medina standard" is based upon the 1971* ***prosecution*** *of U.S. Army Captain* ***Ernest Medina*** *in connection with the My Lai Massacre during the* ***Vietnam War***". This sentence was retrieved with the Triple-Based model from the document "*Command Responsibility*", while the main sentence is from the document "*Ernest Medina*". In Figure 6.3 it is possible to notice a higher volume of visits of the annotated document "*Ernest Medina*".

With the information presented, a new dump was created through the use of Coreference Resolution. For that, the sentences of the document are aggregated. They are linked with a special token *.!* that does not interfere with the performance of the model. Afterwards, the model creates a record of clusters of corefering mentions. Therefore, every corefering mention is replaced with the primary mention in the associated clusters. The inverse process is required in order to remove the special token and maintain the sentence order.

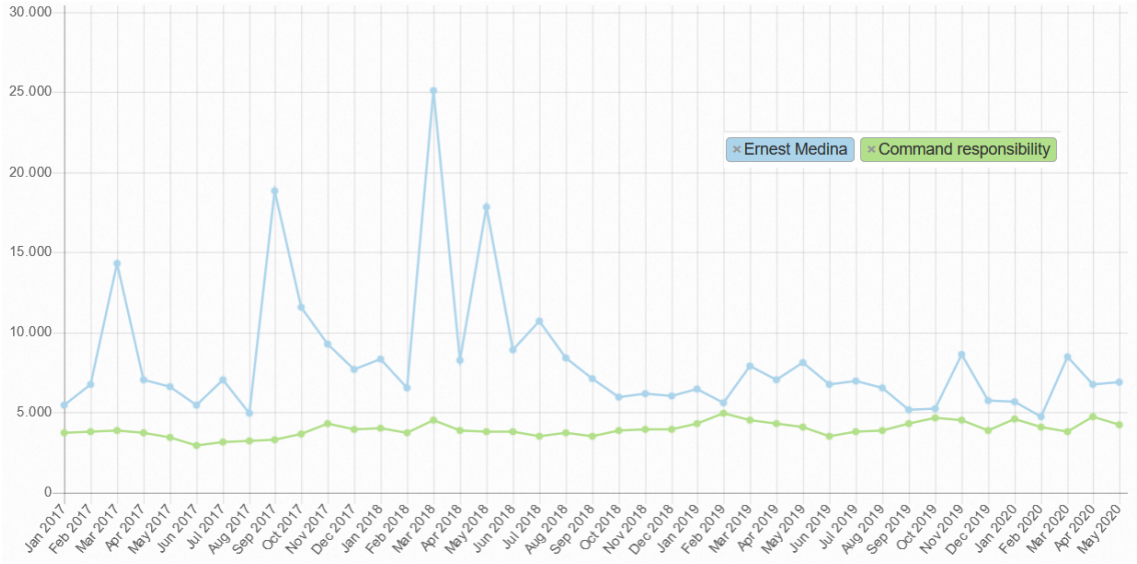

Figure 6.3: A comparison of the number of page views (y axis) between the document "*Ernest Medina*" in blue and "*Command Responsibility*" in green during the period of Jan 2017 and May 2020 (x axis). Source from https://pageviews.toolforge.org

After completing this process for every document, a new dump is created using the pipeline *NeuralCoref 4.0: Coreference Resolution in spaCy with Neural Networks*[1]. For example, in the document "*Portugal*", the sentence "***It** is the 5th most peaceful country in the world, maintaining a unitary semi-presidential republican form of government*" becomes "***Portugal** is the 5th most peaceful country in the world, maintaining a unitary semi-presidential republican form of government*".

Table 6.8: Results of Evidence Retrieval with the Coreference Resolution Wikipedia dump

	Baseline	**Triple-based Model**	**Pairwise Model**	**Pointwise Model**
Precision	0.078	0.069	0.089	**0.105**
Recall	0.699	0.773	0.685	**0.787**
Precision*	0.099	0.075	0.093	0.108
Recall*	0.892	0.807	0.713	0.808

* When at least one group of documents is found

The Triple-Based model was retrained, for the *Evidence Retrieval* task, using the new dump, on 6 189 sentences in a balanced set, extracted from the subsample of the training set. Validating on 1 857 sentences, the new trained TB model achieved an F1-Score of 86.2%, an increase from the previous train (Section 6.1) by 1.5%. The results of the models, in the *Evidence Retrieval* task, using the created Coreference Resolution Wikipedia dump are presented in Table 6.8. The recall of the 3 methods increased: the TB by 1.4%, the pairwise model by 1.8% and the pointwise model by 1.9%. These improvements show that coreference resolution is helpful to give a certain context to

[1] https://github.com/huggingface/neuralcoref

the sentences, better than adding the document title. The precision of only the Transformer-based models increased while the TB model decreased. This decrease was from a Precision of 9.1% to 6.9% representing a decrease ratio of 0.76. Therefore, although the TB model has improved the Recall, more sentences are considered as relevant to the claim showing that more sentences contain claim entities. Nevertheless, it harms the TB model, because more sentences containing those entities, the more sentences are identified as relevant by the model. This analysis reveals that one of the most impactful features is whether a given entity of the claim is contained in the sentence.

Table 6.9: Results of Label Classification with the Coreference Resolution Wikipedia dump

	Pointwise Model	**Pointwise Model with Coref**
Precision	0.476	**0.501**
Recall	0.439	**0.442**
FEVER Score	0.439	**0.444**
F1-Score	0.457	**0.470**
Accuracy	0.552	**0.557**

The impact on the *Label Classification* task, presented in Table 6.9, was only evaluated for the Pointwise model. The results improved, as expected, increasing the new FEVER-Score to 44.4%. This reveals that using coreference resolution can improve the overall performance of the system.

6.5 Summary

In this chapter, different techniques were explored to improve the DeFactoNLP system in the *Evidence Retrieval* task and its impact on the *Label Classification* task. Firstly, a Triple-Based model was introduced, in attempt to explore subject, predicate and object relations between candidate sentences. However, it only increased the Precision in the *Label Classification* task keeping the same FEVER-Score. Later, two different architectures were explored: a Pairwise and a Pointwise model, using a Transformer-based architecture (the BERT base model). The Pointwise model outperformed the Pairwise architecture. Compared to the initial state of the DeFactoNLP system, the Pointwise model increases the Recall from 69.9% to 76.8% in the *Evidence Retrieval* task and also increases the FEVER-Score from 40% to 43.9%. Finally, exploring coreference resolution, a new Wikipedia dump was created, where the anaphoras occurring in the wikipedia article are solved. With this new Wikipedia dump, the results of the Pointwise model increased achieving a Recall of 78.7% in the *Evidence Retrieval* task and a FEVER-Score of 44.4% in the *Label Classification* task.

Chapter 7

Conclusions

In the modern world, the need to fight *Fake News* grows exponentially due to the number of fake information that is being concealed continuously on digital outlets (e.g. social networks). The media requires a technological solution to deal with this problem, since addressing this issue using traditional and manual approaches is impractical. This issue happens due to the time required to verify a fact and due to the amount of information that needs to be processed. This way, Fact-checking is becoming a growing research topic that can congregate different research topics, such as: Natural Language Processing, Information Retrieval, Machine Learning, social media, amongst others.

Fact-checking in natural language became more visible through the creation of challenges such as the FEVER challenge [111]. This challenge tries to bring together the best of different research topics in Natural Language Processing such as Question Answering, Information Retrieval and Natural Language Inference in order to create an end-to-end system capable of finding at least one evidence that can be used to prove a given claim. Locating documents is the first step in the search for evidence, as it is where the text passages with relevant information should be presented. The detection of sentences and, above all, checking their stance, enable the preponderance to check all sides of a given statement being one of the first steps of real-life fact-checkers. Any automatic system will not always be correct, so it is essential to complement it with a human check. Hence, the need arises for all stages of a fact-checking system to be available and easily understood by the people who can work with it. Thus, automated fact-checking frameworks are valuable assets for modern society, but the development of such technology is still in the early stages. A critical step to the performance of such frameworks is *Evidence Extraction*, which comprehends the *Document Retrieval* and *Evidence retrieval* sub-tasks. The performance of each step was investigated in the DeFactoNLP system, a framework which participated in the FEVER 1.0 challenge. We investigated and further demonstrated the impact that these early stages have on the overall performance.

To support the scientific contribution of this dissertation, one article was accepted in an international peer-reviewed conference during the dissertation development [3]. Another article is being prepared for a full-paper submission. The paper "Exploring NLP and Information Extraction to Jointly Address Question Generation and Answering", accepted in the *16th International Conference on Artificial Intelligence Applications and Innovations* (AIAI), explored the Open Information Extraction methods, and the algorithms created to find relevant sentences given a certain claim. The second article being prepared for submission will be titled "Towards Better Fact-Checking Systems Through Evidence Extraction Methods", and will explore the approach that achieved state-of-the-art in the *Document Retrieval* task. The use of Coreference Resolution is also presented as a way to improve the *Evidence Retrieval* models and, therefore, the overall systems in *Label Classification*.

7.1 Discussion of the Research Questions

To understand how the DeFactoNLP system works, an ablation study was conducted (Chapter 4). The first step was the study of the *Document Retrieval* task. Through the defined metrics, it was possible to detect a significant flaw in the NER pipeline, where it did not retrieved any document for about 10% of the dataset. Also, the two techniques used (TF-IDF and NER) analyse different aspects of the documents complementing each other. The TF-IDF extracts documents based on the document's context, and the NER extracts documents based on the document's title. In the second step, *Evidence Retrieval* task, DeFactoNLP system does not present any sentence selection model in the NER pipeline. This causes a discrepancy in precision. Thus, evaluating the end-to-end system and answering the first research question stated in Section 1.2 (*How each fact-checking system component can impact the overall performance?*), it was possible to highlight the need for better precision in the *Evidence Retrieval* task and a good recall in the *Document Retrieval* task. Without a good recall in the *Document Retrieval* task, relevant documents could be missed, affecting the following tasks in the pipeline. Without a good precision in the *Evidence Retrieval* task, the models presented in the *Label Classification* task do not learn correctly, presenting a lower performance. This is because the candidate evidence that is used as input to the *Label Classification* task, creates to much noise lowering the model's performance.

Proceeding to improve the *Document Retrieval* task, Open Information Extraction was used to cover a major flaw in the NER pipeline (i.e., not extracting any document for 10% of the claims). Despite having increased the performance of the system, an error analysis demonstrated the existence of a systematic issue in the system: the inability to retrieve documents that requires a title disambiguation. A novel algorithm was created to retrieve all the documents that may contain those disambiguations (Section 5.2). The algorithm is then followed by a selection process to narrow down those documents, selecting the most significant ones by analysing the document title, the document content and, with the use of external data, the number of documents views (Section 5.3). Since the external data presented such an improvement in the selection of relevant documents, queries were created analysing the claim through Open Information Extraction. These

queries exploit the MediaWiki API that returns the most relevant Wikipedia documents for those given queries (Section 5.4). This method achieved state-of-the-art results, reaching a recall of 90.1% and accuracy of 94.4%. Answering the second research question (*Can Open Information Extraction methods be used to help retrieve documents and possible evidence?*), OIE can indeed help the *Document Retrieval* structuring the claim in a such a way that can be explored to improve the performance of the overall system.

Moving on to the second task, Open Information Extraction was used as an Extraction Method to replace and, therefore, improve the NER pipeline. The precision in the *Evidence Retrieval* task was able to remove 33% of non-relevant sentences (Section 6.1). Furthermore, Transformer-based models were explored using two architectures: pairwise (Section 6.2) and pointwise (Section 6.3). The last architecture provided the highest precision, recall and FEVER-Score. An error analysis detected that, when the correct evidence makes an anaphoric reference to the entity in the claim, the retrieved evidence does not contain the correct evidence because it cannot handle those cases. Therefore, a new Wikipedia dump was created using state-of-the-art Coreference Resolution models. Using the new dump to train the proposed models improved the system's recall in the *Evidence Retrieval* task to 78.7% (an improvement of 8.8% when compared to the DeFactoNLP baseline), while improving the precision in a ratio of 1.35. The FEVER Score was also improved, in the *Label classification* task, to 44.4% (an improvement of 4.44% when compared to the DeFactoNLP baseline).

We also illustrated that the OIE model created for the *Evidence Retrieval* task performs better than the DeFactoNLP system baseline, improving jointly the Precision and the Recall, and thus contributing to better answer the second research question. However, and answering the third research question (*Are transformer-based models, the state-of-the-art in NLP, able to outperform Open Information extraction methods?*), the transformer-based models outperform the OIE model, and only the pipelines using transformer-based models increased the FEVER-Score (hence, the end-to-end system).

7.2 Lessons Taken and Future Work

To develop this dissertation, I had to overcome several challenges. To begin with, understanding a pre-written code in its entirety was a complex task already associated with the complexity of the different pipelines of the system. The document retrieval task requires demanding computational resources due to the large size of the the Wikipedia dump, containing around 5.4 millions documents. Managing the available resources was a challenging task, precluding the creation of complex algorithms, for computer resources purposes, and demanding constant planning on which models to train or not.

In this dissertation, only two of the three tasks were modified. The *Document Retrieval* task achieved state-of-the-art values when compared to the FEVER approaches; in the *Evidence Extraction* the transformer-based models achieved the best results, in specific, the pointwise model. By using Coreference Resolution to solve the anaphoric references, a new Wikipedia dump was

created. This helped improving the models in the *Evidence Retrieval* task. This improvement showed that giving a particular context to the sentence (with solved anaphoric references) is essential for models to better analyse the relation between a claim and a sentence. The *Label Classification* task was not modified in this dissertation, but it is possible to identify pivotal steps which would probably improve this task. For example, using Transformer-based models and also replacing the Random Forest classifier with a state-of-the-art model.

In Section 6.1, the results of the RoBERTa model [65] and BERT model [27] were introduced without fine-tuning. The RoBERTa model, compared to the BERT model, outperforms the model in the *Evidence Retrieval* task, increasing the recall from 51.0% to 63.4%. Hence, it is plausible to speculate that using the presented pointwise architecture (Section 6.3), replacing the fine-tuned BERT model with a fine-tuned RoBERTa [65] or XLNET [125], will increase the overall system performance.

In Section 5.2 and Section 5.3, it was not possible to complete the experiments for the datasets using the presented algorithms that deal with the disambiguated documents. Dealing with this problem can be accomplished by accelerating document search. However, something innovative that was presented was the prospect of using Coreference Resolution, in particular, to provide a specific context to the sentences contained in a given document. This perspective may influence the results, but it has not been explored to the necessary depth to have a more significant impact, since it has only replaced the anaphoric reference by the primary mention in the associated cluster. For example, one could explore how to replace the anaphoric reference by the most similar mention to the claim, or even saving all the mentions and use all of them as possible candidates. Finally, the FEVER dataset uses Wikipedia documents as a basis for finding evidence. An assumption is made that Wikipedia is a verifiable source, although it can be edited by any user. From an application perspective, evaluating whether the source is reliable or not brings a whole new perspective to the problem.

References

[1] Bouziane Abdelghani, Djelloul Bouchiha, Noureddine Doumi, and Mimoun Malki. Question answering systems: Survey and trends. *Procedia Computer Science*, 73:366–375, 12 2015.

[2] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.

[3] Pedro Azevedo, Bernardo Leite, Henrique Lopes Cardoso, Daniel Castro Silva, and Luís Paulo Reis. Exploring nlp and information extraction to jointly address question generation and answering. In Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis, editors, *Artificial Intelligence Applications and Innovations*, pages 396–407, Cham, 2020. Springer International Publishing.

[4] Mevan Babakar and Will Moy. The state of automated factchecking. *Full Fact*, 2016.

[5] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2670–2676, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[6] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[8] 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.

[9] 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.

[10] Eugene Charniak. Statistical techniques for natural language parsing. *AI magazine*, 18(4):33–33, 1997.

[11] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[12] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750, 2014.

[13] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. Enhancing and combining sequential and tree LSTM for natural language inference. *CoRR*, abs/1609.06038, 2016.

[14] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

[15] Noam Chomsky. *Language and mind*. Cambridge University Press, 2006.

[16] Noam Chomsky. *Aspects of the Theory of Syntax*, volume 11. MIT press, 2014.

[17] E.T Chourdakis and J.D. Reiss. Grammar informed sound effect retrieval for soundscape generation. In *DMRN+ 13: Digital Music Research Network One-day Workshop*, page 9, London, UK, November 2018.

[18] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[19] Jennifer Chu-Carroll, Krzysztof Czuba, John Prager, and Abraham Ittycheriah. In question answering, two heads are better than one. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 24–31. Association for Computational Linguistics, 2003.

[20] Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations. *CoRR*, abs/1606.01323, 2016.

[21] Charles LA Clarke, Gordon V Cormack, and Thomas R Lynam. Exploiting redundancy in question answering. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 358–365. ACM, 2001.

[22] Sarah Cohen, Chengkai Li, Jun Yang, and Cong Yu. Computational journalism: A call to arms to database researchers. In *CIDR*, 2011.

[23] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

[24] Lorrie Faith Cranor and Brian A LaMacchia. Spam! *Communications of the ACM*, 41(8):74–83, 1998.

[25] Lei Cui, Furu Wei, and Ming Zhou. Neural open information extraction. *arXiv preprint arXiv:1805.04270*, 2018.

[26] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.

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

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[29] Abdessamad Echihabi, Ulf Hermjakob, Eduard Hovy, Daniel Marcu, Eric Melz, and Deepak Ravichandran. How to select an answer string? In *Advances in open domain question answering*, pages 383–406. Springer, 2008.

[30] Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48, 2016.

[31] Diego Esteves, Anisa Rula, Aniketh Janardhan Reddy, and Jens Lehmann. Toward veracity assessment in rdf knowledge bases: An exploratory analysis. *J. Data and Information Quality*, 9(3), February 2018.

[32] Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings, 2019.

[33] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[34] Mohamed Abdel Fattah and Fuji Ren. Automatic text summarization. *World Academy of Science, Engineering and Technology*, 37:2008, 2008.

[35] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[36] Winthrop Nelson Francis, Henry Kučera, and Andrew W Mackie. *Frequency analysis of English usage: Lexicon and grammar*. Houghton Mifflin Harcourt (HMH), 1982.

[37] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640, 2018.

[38] Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[39] Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. Defacto - temporal and multilingual deep fact validation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2015.

[40] Jeffrey A Gottfried, Bruce W Hardy, Kenneth M Winneg, and Kathleen Hall Jamieson. Did fact checking matter in the 2012 presidential campaign? *American Behavioral Scientist*, 57(11):1558–1567, 2013.

[41] D Graves. Understanding the promise and limits of automated fact-checking. Technical report, University of Oxford, 2018.

[42] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224. ACM, 1961.

[43] Thomas R Gruber. The role of common ontology in achieving sharable, reusable knowledge bases. *KR*, 91:601–602, 1991.

[44] Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. Ukp-athene: Multi-sentence textual entailment for claim verification. *arXiv preprint arXiv:1809.01479*, 2018.

[45] Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. Ukp-athene: Multi-sentence textual entailment for claim verification. *CoRR*, abs/1809.01479, 2018.

[46] Naeemul Hassan, Bill Adair, James T Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. The quest to automate fact-checking. In *Proceedings of the 2015 Computation+ Journalism Symposium*, 2015.

[47] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812. ACM, 2017.

[48] Jerry R Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978.

[49] Ted Honderich. *The Oxford Companion to Philosophy*. Oxford University Press, 2005.

[50] Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question answering in webclopedia. In *TREC*, volume 52, pages 53–56, 2000.

[51] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[52] Tom N Jagatic, Nathaniel A Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[53] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*, 2019.

[54] Ms Anjali Ganesh Jivani, Ms Amisha Hetal Shingala, and Paresh V Virparia. The multiliaison algorithm. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 2(5), 2011.

[55] Vidur Joshi, Matthew E. Peters, and Mark Hopkins. Extending a parser to distant domains using a few dozen partially annotated examples. *CoRR*, abs/1805.06556, 2018.

[56] Mahboob Alam Khalid, Valentin Jijkoun, and Maarten De Rijke. The impact of named entity normalization on information retrieval for question answering. In *European Conference on Information Retrieval*, pages 705–710. Springer, 2008.

[57] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, December 2014.

[58] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[59] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[60] Shalom Lappin and Herbert J Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.

[61] Juan Le, Chunxia Zhang, and Zhendong Niu. Answer extraction based on merging score strategy of hot terms. *Chinese Journal of Electronics*, 25(4):614–620, 2016.

[62] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

[63] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, pages 707–710, 1966.

[64] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

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

[66] Julie Beth Lovins. Development of a stemming algorithm. *Mech. Translat. & Comp. Linguistics*, 11(1-2):22–31, 1968.

[67] John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. Performance measures for information extraction. In *In Proceedings of DARPA Broadcast News Workshop*, pages 249–252, 1999.

[68] Christopher Malon. Team papelo: Transformer networks at fever. *arXiv preprint arXiv:1901.02534*, 2019.

[69] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

[70] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[71] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[72] Mausam Mausam. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 4074–4077, 2016.

[73] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: Can we trust what we rt? In *Proceedings of the first workshop on social media analytics*, pages 71–79. ACM, 2010.

[74] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[75] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[76] Gary Miner, John Elder IV, Andrew Fast, Thomas Hill, Robert Nisbet, and Dursun Delen. *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press, 2012.

[77] Newman Nic, Richard Fletcher, Antonis Kalogeropoulos, David AL Levy, and Rasmus Kleis Nielsen. *Reuters Institute Digital News Report 2018*. Reuters Institute for the Study of Journalism, 2018.

[78] Yixin Nie, Haonan Chen, and Mohit Bansal. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866, 2019.

[79] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, pages 380–384, 2013.

[80] Chris D Paice. Another stemmer. In *ACM Sigir Forum*, pages 56–61. ACM New York, NY, USA, 1990.

[81] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

[82] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[83] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[84] Martin F Porter et al. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[85] John Prager, Jennifer Chu-Carroll, Eric W Brown, and Krzysztof Czuba. Question answering by predictive annotation. In *Advances in Open Domain Question Answering*, pages 307–347. Springer, 2008.

[86] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[87] Jacob Ratkiewicz, Michael D Conover, Mark Meiss, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer Menczer. Detecting and tracking political abuse in social media. In *Fifth international AAAI conference on weblogs and social media*, 2011.

[88] Aniketh Janardhan Reddy, Gil Rocha, and Diego Esteves. Defactonlp: Fact verification using entity recognition, TFIDF vector comparison and decomposable attention. *CoRR*, abs/1809.00509, 2018.

[89] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[90] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[91] Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*, 2017.

[92] Gil Rocha, H Lopes Cardoso, and Jorge Teixeira. Argmine: a framework for argumentation mining. In *Computational Processing of the Portuguese Language-12th International Conference, PROPOR*, pages 13–15, 2016.

[93] Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. Triplet extraction from sentences. In *Proceedings of the 10th International Multiconference" Information Society-IS*, pages 8–12, 2007.

[94] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[95] Edouard Ngor Sarr and Ousmane Sall. Automation of fact-checking: State of the art, obstacles and perspectives. In *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. IEEE, 2017.

[96] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. Open language learning for information extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 523–534. Association for Computational Linguistics, 2012.

[97] Frédérique Segond, Anne Schiller, Gregory Grefenstette, and Jean-Pierre Chanod. An experiment in semantic tagging using hidden markov model tagging. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 1997.

[98] Saeedeh Shekarpour, Edgard Marx, Axel-Cyrille Ngonga Ngomo, and SA Sina. *Semantic interpretation of user queries for question answering on interlinked data*. Elsevier-Web Semantics, 2015.

[99] SimilarWeb. Most visited websites, 1999.

[100] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195 – 197, 1981.

[101] Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, 2013.

[102] Amir Soleimani, Christof Monz, and Marcel Worring. Bert for evidence retrieval and claim verification. In Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, *Advances in Information Retrieval*, pages 359–366, Cham, 2020. Springer International Publishing.

[103] Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, 2016.

[104] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. 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 885–895, 2018.

[105] F. Strier. *Reconstructing Justice: An Agenda for Trial Reform*. University of Chicago Press, 1996.

[106] Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. Anaphora and coreference resolution: A review. *CoRR*, abs/1805.11824, 2018.

[107] Xu Sun. Structure regularization for structured prediction. In *Advances in Neural Information Processing Systems*, pages 2402–2410, 2014.

[108] Scott M Thede and Mary Harper. A second-order hidden markov model for part-of-speech tagging. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 175–182, 1999.

[109] James Thorne and Andreas Vlachos. Adversarial attacks against fact extraction and verification. *CoRR*, abs/1903.05543, 2019.

[110] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. *CoRR*, abs/1803.05355, 2018.

[111] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. The fact extraction and verification (FEVER) shared task. *CoRR*, abs/1811.10971, 2018.

[112] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, and Christina Unger. Hawk–hybrid question answering using linked data. In *European Semantic Web Conference*, pages 353–368. Springer, 2015.

[113] Mike Uschold and Martin King. Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conduction with IJCAI-95*, Montreal, Canada, 1995.

[114] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[115] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[116] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[117] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. A theoretical analysis of NDCG type ranking measures. *CoRR*, abs/1304.6480, 2013.

[118] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[119] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426, 2017.

[120] Nick Wingfield, Mike Isaac, and Katie Benner. Google and facebook take aim at fake news sites. *The New York Times*, 11:12, 2016.

[121] Terry Winograd. Procedures as a representation for data in a computer program for understanding natural language. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC, 1971.

[122] Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. Learning global features for coreference resolution. *CoRR*, abs/1604.03035, 2016.

[123] William A Woods. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pages 441–450. ACM, 1973.

[124] Kun Xu, Sheng Zhang, Yansong Feng, and Dongyan Zhao. Answering natural language questions via phrasal semantic parsing. In *Natural Language Processing and Chinese Computing*, pages 333–344. Springer, 2014.

[125] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019.

[126] Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, 2007.

[127] Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Ucl machine reading group: Four factor framework for fact finding (hexaf). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102, 2018.

[128] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. *SIGIR Forum*, 51(2):268–276, August 2017.

[129] Junlang Zhan and Hai Zhao. Span model for open information extraction on accurate corpus, 2019.

[130] Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. Reasoning over semantic-level graph for fact checking, 2019.