UNIVERSITÀ
DEGLI STUDI
FIRENZE

PhD Program in Smart Computing
Dipartimento di Ingegneria dell'Informazione (DINFO)

Combining Natural Language Processing and Machine Learning for Profiling and Fake News Detection

Alessandro Bondielli

Dissertation presented in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Smart Computing

PhD Program in Smart Computing
University of Florence, University of Pisa, University of Siena

Combining Natural Language Processing and Machine Learning for Profiling and Fake News Detection

Alessandro Bondielli

Advisor:

Prof. Francesco Marcelloni

Head of the PhD Program:

Prof. Paolo Frasconi

Evaluation Committee:
Prof. Alberto Bartoli, *University of Trieste*
Prof. Roberto Basili, *University of Roma Tor Vergata*

XXXIII ciclo — June 2021

Acknowledgments

Throughout my PhD and the writing of this thesis I received a great deal of support and assistance, for which I will be forever grateful.

First and foremost, I would like to thank my advisor Professor Francesco Marcelloni, for his invaluable help and guidance during my PhD. He has been an amazing advisor under many different lights, and his insights helped me to grow as a researcher. I am also truly thankful to Prof. Alessandro Lenci, who pushed me to pursue this path, for his teachings, and all the help he provided me with his invaluable advice.

I would like to acknowledge also my Supervisory Committee, formed by Professors Beatrice Lazzerini, Francesco Marcelloni, Alessandro Lenci and Alessio Bechini. Each year their suggestions and feedback were very valuable to find the right direction in continuing my research.

I thank my Evaluaton Committe, Prof. Alberto Bartoli and Prof. Roberto Basili, for their review of my work and their helpful comments also during the preliminary defense. Their inputs undoubtedly improved these pages.

I also would like to thank all my collaborators and colleagues both from the engineering department and the CoLingLab, which helped in making this experience fruitful and enjoyable. I especially want to single out Lucia Passaro, for being the first to believe in me as a researcher, for being an amazing and fundamental support trough all the stages of my research and for sharing with me both the joys and and the sorrows of this journey. Last but not least, for becoming much more than a colleague, one of the most important people my life.

Part of the present work was developed during my involvement in the projects TALENT 4.0, funded by Regione Toscana. I would like to thank all the partners of the project that enabled me to see my research work applied into an industrial context. In addition, other aspects of this thesis were developed thanks to the involvement in the MIT-UNIPI program.

Finally, I must not forget to thank my friends and family, as well as all the other important people in my life. My parents, for always being there, never failing to support and believe in me. The rest of my family for making me feel loved. My friends, for standing by me and sharing all the best and worst moments, and always being able to put a smile on my face when I needed it most. My girlfriend Ottavia, who could see only the final miles of this journey but in that little (*for now*) time never failed in supporting and enduring me. Every other important person in my life, some more than others. All of them have always been there for me, some even more than I ever hoped for, and without them, their love and care and support, I could never have made it here today. To them, the most heartfelt thanks.

Abstract

In recent years, Natural Language Processing (NLP) and Text Mining have become an ever-increasing field of research, also due to the advancements of Deep Learning and Language Models that allow tackling several interesting and novel problems in different application domains. Traditional techniques of text mining mostly relied on structured data to design machine learning algorithms. Nonetheless, a growing number of online platforms contain a lot of unstructured information that represent a great value for both Industry, especially in the context of Industry 4.0, and Public Administration services, e.g. for smart cities. This holds especially true in the context of social media, where the production of user-generated data is rapidly growing. Such data can be exploited with great benefits for several purposes, including profiling, information extraction, and classification. User-generated texts can in fact provide crucial insight into their interests and their skills and mindset, and can enable the comprehension of wider phenomena such as how information is spread through the internet.

The goal of the present work is twofold. Firstly, several case studies are provided to demonstrate how a mixture of NLP and Text Mining approaches, and in particular the notion of distributional semantics, can be successfully exploited to model different kinds of profiles that are purely based on the provided unstructured textual information. First, city areas are profiled exploiting newspaper articles by means of word embeddings and clustering to categorize them based on their tags. Second, experiments are performed using distributional representations (aka embeddings) of entire sequences of texts. Several techniques, including traditional methods and Language Models, aimed at profiling professional figures based on their résumés are proposed and evaluated. Secondly, such key concepts and insights are applied to the challenging and open task of fake news detection and fact-checking, in order to build models capable of distinguishing between trustworthy and not trustworthy information. The proposed method exploits the semantic similarity of texts. An architecture exploiting state-of-the-art language models for semantic textual similarity and classification is proposed to perform fact-checking. The approach is evaluated against real world data containing fake news. To collect and label the data, a methodology is proposed that is able to include both real/fake news and a ground truth. The framework has been exploited to face the problems of data collection and annotation of fake news, also by exploiting fact-checking techniques.

In light of the obtained results, advantages and shortcomings of approaches based on distributional text embeddings are discussed, as is the effectiveness of the proposed system for detecting fake news exploiting factually correct information. The proposed method is shown to be a viable alternative to perform fake news detection with respect to a traditional classification-based approach.

Contents

Chapter 1

Introduction

The last years have marked a fundamental shift in how information is spread across the world and how such information can be accessed, modelled and exploited for various purposes. On the one hand, the *Internet of Things* has led to the introduction of the concepts of *Industry 4.0*, i.e. the automation of various aspects of the industrial processes. On the other hand, novel ways to connect people enabled the creation of new channels of information and communication that can provide different kinds of knowledge.

In this context, language technologies are becoming more and more relevant, both from a research perspective and from an industrial standpoint. Researches related to *Natural Language Processing* (NLP) and *text mining* have in fact seen a steady growth in the last years, both in several published research and advancements in the field. Concerning the latter point, arguably the advancements of Deep Learning frameworks and the introduction of effective and efficient *Neural Language Models* (LM), from earliest approaches such as *word embeddings models* like *word2vec* (Mikolov et al., 2013a,b) to the most recent ones based on the *Transformer architecture* (Vaswani et al., 2017), have allowed to face several different and novel challenges both from the point of view of Computational Linguistics and from an Information Engineering perspective. Traditional techniques of text mining mostly rely on the exploitation of structured data for designing learning algorithms for specific tasks. From a linguistic perspective these data include for example structural patterns of texts and the frequency of words or the inclusion of specific words. Thus, feature modelling is crucial for many text mining approaches (e.g. clustering, classification), in order to find the set of features, descriptive or textual, that best describe the problem. On the contrary, one of the key aspects of neural language modelling lies in the ability to encode and model natural language (i.e. plain texts, with little to no additional information) in order to provide the same, if not better, predictive capability. Especially within the context of the deep learning revolution, aspects of feature modelling have assumed a position of minor importance with respect to

issues related to the architecture of models themselves.

Arguably, when considering analyses that focus on the properties of texts, one of the most crucial aspects, also with respect to more traditional feature-oriented approaches to text mining and NLP, is the modelling of semantics. In fact, one of the driving aspects of research related to language modelling is exactly the idea of obtaining a machine readable representation of texts, as single words or entire sequences, that is able to encode also their meaning. This allows in turn facing more complex problems that require a more complex understanding of text semantics. Actually, the currently dominant approach to computational meaning representation is based on the so-called distributional hypothesis (Harris, 1954; Firth, 1957), which states that word usage in context is directly related to word meaning. Thus, words that share similar contexts of use also tend to share similarities in meaning. The distributional hypothesis has been applied in different forms to most language models concerning the semantics (see Section 2.1). Such language models have been at the forefront of a wide portion of NLP research, and has been proven extremely effective in facing diverse problems that relate to the semantics of words.

As social media become more and more pervasive in everyday life, the availability of unstructured, user-generated information is steadily growing especially thanks to the current state of the Internet. The ability of representing and exploiting such knowledge, including the meaning of words or sequences of words, definitely represents an added value that can benefit both companies, especially considering the ideas behind Industry 4.0, and services such as Public Administrations to improve the quality of life. This knowledge can in fact provide insights for example into profiling problems of various nature, information extraction, and the semantic classification of texts. Companies can, for example, benefit from the ability to predict how the market is set to change, how its current behaviour can influence their business strategies, and how to understand their customers and vendors in order to provide better services. On the other hand, Public Administrations can for example exploit different aspects of social media data in order to better communicate with citizens and to obtain feedback on the perception of the various aspects of the life in the city, and provide strategies to improve them. Moreover, such data can be exploited to analyse linguistics and social phenomena such as the spread of unverified information, and more generally of how information is propagated through the population. In this regard, one aspect that is of interest for the present work is the phenomenon of disinformation and fake news. It is surely becoming one of the most widely debated problems both from a social and a research perspective. The ever-growing proliferation of fake news, and more generally of disinformation and misinformation is strictly related to social media, as they are one of the most fertile grounds for disinformation (Zubiaga et al., 2018a), thanks to the lack of control, at least within certain limits, on the content produced by users. Many current

approaches exploit either surface-level features (e.g. the presence of lexical and syntactical markers), propagation-level features (i.e. how the news is spread on social media and who are the users that propagate it) or a mixture of them in order to identify fake and real news. However, in order to model the problem of fake news, it is crucial to be able to model and understand their meaning in relation to their context, for example represented by trustworthy news. In fact, it is often the case that fake news, rather than being completely false claims, are the product of slight modification of real world events in order to push a specific narrative in the minds of the readers. In these cases, the fake news is much harder to identify as the events depicted in it are both believable and lean on real world events in order to appear more realistic. For example, a real world event can be described by distorting several key facts that may lead the public opinion towards the direction pushed by the authors of the fake news, that have often political or social motivations. Therefore, an analysis that focuses more on the meaning of words and sentences, and that can exploit the relationship between what is actually stated in the real and fake news to distinguish them may provide novel insight and a better modelling of the problem, which can be generalized also to real case scenarios.

1.1 Goals and contributions

The present work focuses on aspects of semantics related to two main areas of application, namely profiling and fake news detection. It can be considered as having two main goals. The first main goal is to understand how text and data mining problems can be faced from a distributional semantic perspective. Specifically, the problem of profiling in several real-world applications is tackled with a mixture of NLP and text mining approaches, with a particular focus on the evaluation of distributional semantics methods. The second main goal is to evaluate how the concepts of distributional semantics can be applied to approach the problem of fake news detection. While traditional systems rely on less semantically oriented features, the approach proposed in this thesis is instead entirely focused on the identification of real and fake news based on the meaning of news articles with respect to a verified, and trustworthy, ground truth.

Profiling

As for the first main goal, the objective is to understand the effectiveness of distributional semantic models in real-case scenarios, and their advantages and drawbacks with respect to more traditional text mining approaches. Moreover, it is interesting to notice that comparisons are made between available *pre-trained* distributional semantic models and models built from the ground up based on the available data.

Two case studies are presented to this end, that trace back to as many applications in the context of Smart Cities and Industry 4.0.

Profiling for smart cities. In the first case study, pre-trained word embeddings and their properties are taken into account in order to extend a smart city framework aimed at profiling the areas of a city with respect to different aspects. Specifically, the goal is to provide an additional profiling of the areas of the city based on news found in online sources. The proposed system exploits pre-trained word embeddings in order to generate clusters of tags of news articles with a similar meaning. Clusters of semantically similar tags are then associated with a macro-category, that can be used in turn to label the news articles. Finally, an SVM classifier is trained to label new articles with their macro-category. The system is evaluated on several years of data concerning the city of Rome. The proposed system is then integrated within the framework in order to profile areas of the city with respect to the macro-categories by means of geo-localization.

Profiling résumés. In the second case study, the problem of profiling professional figures based on their résumé is explored. As the job market is increasingly more dynamic, especially for the sectors most influenced by the ideas of Industry 4.0, it becomes paramount to enable effective and efficient strategies to identify profiles of workers. Most traditional systems focus on small sets of hand-crafted features and simple rule-based algorithms, in conjunction with expert knowledge. This is costly both in time and resources. In order to avoid this problem, one possible strategy is to focus directly on résumés. Therefore, two different approaches to profile professional figures based only on their résumé are proposed. In the first one, several traditional NLP techniques are used to extract keywords, and a doc2vec model (Le and Mikolov, 2014) based on the keywords is trained on the available data in order to obtain distributed representation of entire résumés. Then, profiles are identified by means of clustering on such representation. In the second one, the same goal is faced from a pre-trained perspective, by means of implementing a summarization algorithm to shorten résumés and the state-of-the-art Transformer-based architecture proposed in Reimers and Gurevych (2019) in order to obtain résumé embeddings. As for the first method, profiles are then identified by means of clustering. Both approaches are evaluated qualitatively and quantitatively, with the Transformer-based one obtaining the most promising results.

Fake News detection

As for the second main goal, namely the application of distributional semantics to the problem of fake news detection, three different aspects are considered. First, the

study and implementation of a system focused on performing fact-checking of statements based on verified claims. The core of the system is represented by a Sentence-Transformer model (Reimers and Gurevych, 2019), fine-tuned on two tasks in order to classify whether for pairs of sentences one actually verifies the other one. Second, a methodology to collect and label real and fake news and a ground truth for real-world events is proposed and applied to the Notre Dame fire of 2019. The method collects reliable news and potentially false ones based on tweets, and labels them via crowdsourcing. Finally, the fact-checking methodology is adapted to the task of fake news detection and evaluated on the collected dataset. The proposed method is shown to be viable both in the context of fact-checking and as an interesting future direction for fake news detection, proving that semantics plays a crucial role in understanding how fake news actually configure themselves with respect to real ones.

Contributions

The contributions of the present work can be summarized as follows.

- Evaluation of distributional semantic models, with particular attention to pre-trained ones, in different application contexts. Such systems are evaluated across several profiling tasks, with varying degree of complexity with respect to current approaches.
- Development of a framework for profiling city areas based on newspaper information.
- Proposal and evaluation of several alternatives to face the problem of profiling professional figures based on their résumés.
- Development of a fact-checking system that obtains very good performances on a benchmark dataset.
- Development of a methodology for data collection in the realm of fake news.
- Proposal of an approach for fake news detection based on distributional semantics on the one hand, and on the concept of fact-checking on the other hand.

1.2 Structure of this thesis

This thesis is organized as follows. In Chapter 2 a thorough literature review will be conducted on the two main aspects of interest of the present work, namely the

theory and evolution of distributional semantic models (Section 2.1), and several aspects of fake news detection (Section 2.2).

Chapter 3 presents an approach to the profiling of city areas based on information reported in online news. The approach focuses on two aspects. On the one hand, pre-trained word embeddings of tags associated with each article are exploited to identify several macro-categories of news through clustering analysis. On the other hand, macro-categories are used to label news articles and train a text categorization model based on an SVM classifier to label articles with the respective macro-category. The approach is tested by profiling the city areas of Rome, considering articles from 2014 to 2018.

In Chapter 4 several methods for obtaining distributed representations of text sequences are evaluated in the context of profiling résumés. Specifically, a baseline representation based on fastText word embeddings (Bojanowski et al., 2017), doc2vec (Le and Mikolov, 2014), and Sentence-BERT (Reimers and Gurevych, 2019) is evaluated. In all cases, the distributional semantic models, in conjunction with NLP techniques such as keyword extraction and summarization, are used as feature extractors for résumés. Then, hierarchical clustering is applied to résumé embeddings to identify professional figures profiles. Results are evaluated both qualitatively and quantitatively.

Chapter 5 tackles the problem of fake news detection. In particular, first an approach to perform fact-checking and verified claim retrieval is described. The system is based on Sentence-BERT, to which two cascade fine-tuning steps are applied in order to identify claims that verify tweets. Second, a methodology for collecting and labelling data containing real and fake news for a specific event is proposed, and a case study on the Notre Dame fire of April 2019 is presented. Data concerning the fire are collected, from both reliable and unreliable sources, and labelled via crowdsourcing. Third, the fact-checking model is adapted and applied to the task of fake news detection on the Notre Dame fire dataset. Obtained results are discussed with specific focus on the properties of the data collection strategy and on the adaptation of the fact-checking system.

Finally, Chapter 6 provides an overview of the findings and insights obtained from the performed experiments. Chapter 7 draws some conclusions.

Chapter 2

Literature Review: Textual Similarity and Fake News

In the last few years NLP and Computational Linguistics have experienced an exponential growth of interest from several different areas and domains of application. This growing interest is motivated by substantial advancements in the field, both in downstream applications such as text classification and categorization, machine translation, and natural language generation among others, and in the development of novel techniques based on machine and deep learning to represent the semantic information of words and entire texts in an efficient and effective machine readable format. In fact, most of such applications rely on the semantics of words and sentences to improve their performances.

In this Chapter, two key aspects will be discussed. On the one hand, the evolution of computational methods for modelling word and sentence meaning, in order to enable the computation of textual similarities. On the other hand, a specific downstream task will be taken into account, namely the fake news and rumour detection one, as Chapter 5 specifically tackles the problem with the help of state-of-the-art methods for understanding the semantics of texts.

2.1 Textual and semantic similarity from a computational perspective

Modelling the semantics of texts is one of the most widely studied aspects in Natural Language Processing and Computational Linguistics. In fact, obtaining human-level understanding of the meaning of words and sentences is a crucial aspect in many downstream applications of NLP, such as text categorization, question answering and machine translation among others. In all such applications and many others, it is fundamental to have a way in which to measure semantic similarity and

relatedness of words and sentences, thus deriving machine understandable representations of the meaning of words and concepts, and how they actually relate to each other.

As the estimation of the similarity between words has been a very active field since the early days of Computational Linguistics and NLP, two main families of methods and metrics can be identified, namely *ontology-based* and *distributional-based* ones (Lastra-Díaz et al., 2019).

A short review on ontology-based methods

While the focus of the present research is heavily shifted towards distributional representations of meaning, it is nonetheless interesting and important to at least provide some basic concepts regarding ontology-based representations of meaning. A clear definition of ontology is not straightforward, as it depends on several factors such as what is represented in the ontology and how. A number of different definitions have been proposed in the literature. However, one key aspect they share is the fact that all ontologies aim at representing some kind of entities (e.g. concepts, real-world objects, or both) and the relations among them in a specific domain. Typically, the most basic form of relationship among concepts represented trough ontologies is that of "is-a". For example, a *car* is a *vehicle*. Clearly, other semantic relations can be encoded, such as for example hypernymy, hyponymy, meronymy, antonymy, and synonymy (Lastra-Díaz et al., 2019).

One of the most well regarded ontologies is WordNet (Miller, 1995; Fellbaum, 1998). WordNet is a hand-crafted ontology that encodes several different types of relationship among concepts. It exploits the idea of cognitive synonyms, or *synsets*, to express a concept represented as either a noun, verb, adjective or adverb. Synsets are linked to each other by means of semantic and lexical relations (Fellbaum, 2005). WordNet includes around 117,000 English synset, and several different relations among them, based on their Part-of-Speech (PoS). Specifically, nouns are connected through hyponymy/hyperonimy and meronymy relations, while verbs are organized by troponymy, and adjectives in terms of antonymy. Several efforts to exploit the categorization of WordNet in other languages have been performed, such as for example MultiWordNet (Pianta et al., 2002).

In order to compute the similarity (or relatedness) of concepts organized as such, several different measures have been proposed in the literature. Lastra-Díaz et al. (2019) propose the categorization of ontology-based similarity measures in *semantic topological measures*, *gloss-based measures*, and *vector representation models*.

Semantic Topological Measures exploit only the topology of the ontology to derive similarity and distance among concepts, by considering indexes such as the path from one concept to another, the similarity of their feature and so on. Examples are path-based similarity measures (Rada et al., 1989; Dong et al., 2010), information

content-based similarity measures (Resnik, 1995; Lin, 1998b), feature-based measures (Tversky, 1977; Likavec et al., 2019), and graph-based measures (Stanchev, 2014; Quintero et al., 2019). On the other hand, Gloss-based models hinge on the idea that semantically similar concepts or words are described with similar glosses in the ontology. Therefore, most approaches focus on a measure of the similarity of glosses, in terms of shared words, embedding vectors or weighting measures for shared nouns (Lesk, 1986; Banerjee and Pedersen, 2003; Patwardhan, 2006; Aouicha and Taieb, 2015). Finally, with vector representation models, words and concepts are typically compared by considering their representation based on the graph structure of the ontology (Agirre and Soroa, 2009; Goikoetxea et al., 2015; Camacho-Collados et al., 2016).

Distributional semantics: key concepts and applications

While ontologies are often hand-crafted, and refer to specific domains of application in order to define similarity and relatedness among words, Distributional Semantics relies instead on the distribution of words to model their meaning, starting from the assumption that an important role in the semantics of words is played by how they are distributed in context (Lenci, 2018).

All the literature concerning distributional semantics stems from the *Distributional Hypothesis*, first introduced in Harris (1954) and Firth (1957): words that share similar linguistic contexts tend to share also similar meanings. Such hypothesis has been extensively explored throughout the years, and researches have proven how actually the contextual information is a good approximation of the word meaning (Miller and Charles, 1991).

Models of meaning based on the distributional hypothesis therefore share the common characteristic of representing a word, and thus its meaning, based on its context, typically obtained from corpora. In most distributional semantic models, words are represented as n-dimensional vectors which encode their contexts in corpora, as a proxy of their meaning representation (Baroni et al., 2014). Several different techniques to obtain such representation have been proposed in the literature. Baroni et al. (2014) argues that two main paradigms to the creation of Distributional Semantic Models have been studied, namely the *count-based* and *prediction-based* ones.

Traditional count-based models

In count-based models, which account for the earliest approaches to the problem, the key idea is to simply encode and count the occurrences of a particular word in a particular context as feature, i.e. one of the n dimensions of the word vector. This is typically obtained trough the creation of a *word-context* matrix. In the simplest

cases, the contexts can be considered as the documents in the collection. Let $D = d_1, ..., d_n$ be a collection of documents containing n texts, and $W = w_1, ..., w_m$ be the m unique words found in the collection D. The term-document matrix X is therefore a matrix with m rows, one for each word, and n columns, one for each document. X_{ij} represents the number of occurrences of the word w_i in the document d_j. Thus, each row is actually an n-dimensional vector that represents the word in terms of its frequency in each document in the collection. Conversely, each column is an m-dimensional vector that represents the document in terms of the words that compose it. One of the pioneering works in this regard is that of Salton et al. (1975), where the authors employ the document representation obtained by the vector space model for automatic indexing in search engines. Such approach, despite its simplicity, has been and still is a widely popular method to represent documents in terms of their words, also for considering them as features for machine learning algorithms, and it is mostly known as the *bag-of-words* (BOW) representation.

For more accurate representations of the similarities among words rather than among documents, the considered context may differ from the whole document. For example, words within a window of the target word, phrases, sentences or other syntactical constructions may be taken into account to act as the context, depending on the goal of the analysis (Lund and Burgess, 1996; Lin, 1998a; Padó and Lapata, 2007; Erk and Padó, 2008). In this case, the similarity of word vectors (i.e. the rows of the word-context matrix) is used to derive the similarity of word meanings (Turney and Pantel, 2010).

As whole documents may not be ideal to be considered as relevant context, also raw frequencies may not represent the best option for describing the features of a word. In fact, a number of researches addressed exactly this problem, by proposing several different weighting techniques that would enable a better representation. Weighted word vectors have in fact been proven to perform better that raw frequencies (Baroni et al., 2014). Most weighting techniques are based on assumptions from information theory, such that most information is carried by *surprising* events rather than expected ones (Shannon, 1948). Classical and widely used examples of weighting schemes are tf-idf (term frequency x inverse document frequency) (Sparck Jones, 1988; Salton and Buckley, 1988), and Pointwise Mutual Information (PMI) (Church and Hanks, 1989; Turney, 2001), or its variation known as Positive PMI (PPMI) (Niwa and Nitta, 1994). Both these weighting schemes have been introduced in the context of information retrieval, but have been successfully applied to also language modelling (Bullinaria and Levy, 2007; Turney, 2008).

Finally, another noteworthy aspect concerning traditional techniques of word representation is the sparsity of representation. In fact, usually word-context matrices are very sparse. This means that most elements of the matrix are actually zeros, since any word will appear in only a small fraction of all the possible con-

texts derived from the data. In order to reduce the dimension of the matrix, and thus reduce the computational cost of comparing vectors, several approaches have been proposed in the literature. One of the earliest approaches is *Truncated Singular Value Decomposition* (Truncated SVD) (Deerwester et al., 1990). Given a word-context matrix X of rank r, the goal is to obtain a matrix of rank k that minimizes the approximation error with respect to X. This is achieved by performing several linear algebraic operations on the matrix, such as decomposition. The approach and its variants have been rather popular in the literature concerning vector space models, such as Landauer and Dumais (1997); Rapp (2003); Brand (2006); Gorrell (2006); Harshman (1970). As argued in Turney and Pantel (2010), in addition to Truncated SVD, several alternative processes for dimensionality reduction of matrices have been proposed, such as Nonnegative Matrix Factorization (NMF) (Lee and Seung, 1999), Probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999), Iterative Scaling (IS) (Ando, 2000), Kernel Principal Components Analysis (KPCA) (Schölkopf et al., 1997), Latent Dirichlet Allocation (LDA) (Blei et al., 2003), and Discrete Component Analysis (DCA) (Buntine and Jakulin, 2006).

Modern prediction-based neural language models

Instead of relying on a representation based on word-context frequency and heuristics for building, weighting and transforming such vectors, most modern approaches, referred to as *neural language models* (Bengio et al., 2003), exploit supervised learning to obtain n-dimensional representations of words. The basic assumption is that the probability of a given word can be estimated by the words that surround it. In its simplest form, given a sentence $S = w_1, ..., w_n$, the probability of the sentence can simply be expressed as the product of the probability of each word in the sentence (Wang et al., 2019):

$$p(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} p(w_i)$$

At the same time, the probability of the nth word can be estimated as the conditional probability of the word given the previous n words in the sentences, as $p(w_i) = p(w_i|w_{i-n+1}, w_{i-n+2}, ..., w_{i-1})$. Thus, in neural language models, the estimation problem is treated as a supervised learning tasks, where the weights in an n-dimensional vector actually maximize the probability of the context in which the word is found (Baroni et al., 2014). This strategy has been for example employed in earliest and pivotal works on neural language models, such as Collobert et al. (2011); Bengio et al. (2003); Mikolov et al. (2013a,b).

The neural language models have several main advantages over count-based ones. First, by exploiting a supervised approach, they employ a clear training objective, that can be suited to different uses, and avoid any direct use of heuristics for

modelling the vectors. Second, albeit the learning approach is supervised in nature, no manual labelling of data is needed. In fact, training data can be automatically obtained from non-annotated corpora, given a value for the context window. Third, the resulting vectors are dense. This is because the length of the resulting vector can be manually chosen before training, and each dimension encodes several different properties. This in turn avoids any further loss of information due to dimensionality reduction techniques. In addition to this, given the fixed length of the vectors, such learning schemes are able to scale better to huge collections of data, as no co-occurrence matrices are needed. In fact, most of such systems are trained on billions of tokens. Fourth, learning algorithms and especially the most novel ones based on deep learning, such as for example BERT (Devlin et al., 2019), can fully exploit the computational power of GPUs to speed up the computation. Finally, trained models can be used in two ways. On the one hand, it is often the case that the same architecture used to build the model can be exploited also for downstream NLP tasks, thus avoiding the need for feature engineering typical of traditional machine learning algorithms. On the other hand, once the vectors for words in the vocabulary are learned during training on a specific corpus, the learned model can be used to extract vectors of words in any other context without re-training. This operation is typically called pre-training of language models.

It could be argued that 2 different approaches can be identified for the creation of neural language models: *non-contextualized* and *contextualized language models*. In non-contextualized language models, or word embedding models, shallow neural networks are typically used to compute a single vector for each word type. This means that results can be stored as $m \times n$ matrices, where m is the size of the vocabulary and n is the number of dimensions chosen for the resulting embedding. Non-contextualized models will be the object of analysis in Section 2.1.

Contextualized models exploit more complex and deeper architectures, such as bi-directional LSTM, encoder-decoder networks, and Transformers, to obtain a contextualized representation of words and sentences. Such models are born from the necessity to model also the information of the context. They are typically trained with specific learning paradigms that allow for a representation of entire sequences of words rather than single words, and can be often fine-tuned after training in order to solve also downstream tasks. Once the language model is trained, it can predict the embeddings of entire sequences of words, and of each single word token in the sequence. Crucially, the final output for each word will be influenced by its surrounding context. For example, in the sentences "Apple is releasing the new MacBook Air" and "The apple does not fall far from the tree", the word "apple" will have two different vector representations. In this case, embeddings are not stored, but rather obtained as the predicted output of passing the entire sequence through the learned model. Contextualized language models will be thoroughly analyzed

in section 2.1.

Using Neural Networks to train Word Embedding Models

The earliest attempt to generate a distributed representation of words trough neural networks was proposed in Bengio et al. (2003), with an architecture appropriately named *Neural Network Language model* (NNLM). Specifically, NNLM generates the embedding by learning to predict the next word given the previous words in the sentence. The architecture is a three-layer neural network, that takes as input the feature vectors (i.e. the embeddings) of n previous words, and learns to predict the feature vector for the subsequent word. Clearly, despite the learning being supervised, no labelling is needed as the only thing the algorithm considers is the sequence of words. First, the inputs are mapped to a conditional probability distribution over the vocabulary of words. Then, tanh is applied to predict the next word. The algorithm uses *backpropagation* to update the weights of the network, and thus of the distributed representation of the words.

Arguably, the most influential of the earliest models for learning word embeddings is Word2Vec (Mikolov et al., 2013a,b). Authors propose two different algorithms that are based on the same general idea that the whole context plays an important role in the prediction of the probability for a target word. Therefore, instead of relying only on the previous words to predict the next one, authors propose to use a window of words surrounding the target one. The two proposed algorithms, namely CBOW (which stands for Continuous Bag-of-Words) and Skip-gram, face the problem from opposite perspectives. While the training objective of CBOW is to learn representation of words in the context window that can best predict the target word, Skip-gram reverses the paradigm by trying to learn representation of target words that can be used to predict the surrounding context. The architecture of CBOW is very similar to the NNLM one (Bengio et al., 2003), since it consists of an input, a hidden, and an output layer. The input layer accepts n-dimensional embeddings for the words in the context. Then, such words are averaged in the hidden, or projection, layer. The *Bag-of-Words* part of the name comes exactly from the fact that, as the context words representation are averaged to obtain the target word representation, their order in the sentence is not taken into account. Finally, such representation is used to predict the target word. Backpropagation is used to adjust the feature vectors of each word in the context to maximize the probability of predicting the target word.

Conversely, the Skip-gram model starts from the target word, and the goal is to maximize the probability of predicting another word in the same sentence, i.e. in its context. Specifically, the current word is used to predict words within a certain range around the word itself via a log-linear classifier. Again, backpropagation is used to adjust the weights. Authors note that, as the context size increases, the vector repre-

sentation improves. However, since more distant words are less relevant to describe the current word, authors propose to under-sample such distant words during the generation of training examples (Mikolov et al., 2013a). A visual representation of CBOW and Skip-gram is shown in Figure 2.1.

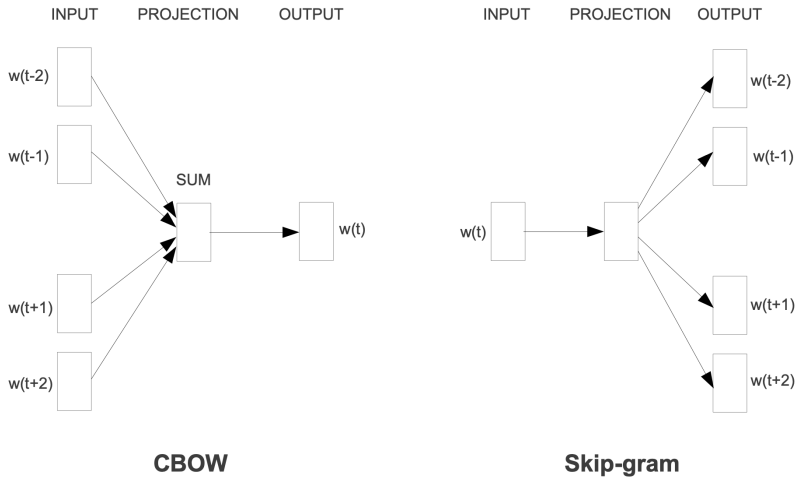


Figure 2.1: CBOW and Skip-gram models proposed in Mikolov et al. (2013a).

In Pennington et al. (2014), authors argue that one of the key drawbacks of the CBOW and Skip-gram family of algorithms is that they do not leverage the global corpus statistics during training. In fact, in Word2Vec models, embeddings are trained only on the local context of words, i.e. the rather small context window around the current word. Thus, authors propose GloVe (Pennington et al., 2014), a method for deriving word embeddings from a global co-occurrence matrix. In GloVe, first the relationship between two words is established as the ratio between the co-occurrency of such words with a set of *probe* words. Given two words i and j, and a probe word k that co-occurs with both of them, the ratio of $\frac{P_{ik}}{P_{jk}}$ is very small if k is related to j but not to i, very high if it is related to i but not to j, and is close to 1 if it is related or unrelated to both. With this in mind, authors propose log-linear function to approximate such relationship between words as:

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_j = \log(X_{ij}) \tag{2.1}$$

where X_{ij} is the co-occurrence frequency of i and j in the corpus, w and $\tilde{w}$ are word vectors and context vectors respectively. The two b are bias terms.

Then, authors propose a weighted least square regression model with a weighting function $f(X_{ij})$ to learn the embeddings, such that:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_k + b_i + \tilde{b}_j - \log(X_{ij}))^2 \tag{2.2}$$

where V is the vocabulary. Finally, considering x_{max} as the maximum number of co-occurrences for a ij pair, the empirically chosen weighting function is the following:

$$f(x) = \begin{cases} (x/x_{max})^{0.75} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases} \tag{2.3}$$

GloVe was shown to perform better than both CBOW and Skip-gram on a number of word analogy tasks.

One key issue with both GloVe (Pennington et al., 2014) and Word2vec (Mikolov et al., 2013a,b) is that they are limited by the words included in the vocabulary V of the chosen training corpus. Therefore, embeddings for out-of-vocabulary (OOV) words cannot be obtained. This is crucial both when the vocabulary is limited in the case of low-resource languages, and when dealing with domain-specific words. Several models have been proposed to address this issue. Arguably, the most popular among such models is fastText (Bojanowski et al., 2017; Joulin et al., 2017). In fastText, authors propose the use of sub-word information for learning embeddings in a Skip-gram model. Specifically, authors propose to encode words by considering their character n-grams, in addition to the word itself, using a special notation. For example, the word *where* is represented by both <where> (note the special characters to describe word boundaries) and <wh, whe, her, ere, re> for character 2- and 3-grams. Note also that the n-gram *her* is different from the word <her> (Bojanowski et al., 2017). Finally, the word is represented as the sum of the vector representations of its n-grams. The proposed method is shown to outperform both models that do not take into account sub-word information and methods that rely on the morphological analysis (Bojanowski et al., 2017), and performs well also on the similarity between in-vocabulary and out-of-vocabulary words thanks to the sub-word model.

Context-aware Language Models

All the previously mentioned models and algorithms, both count-based and predict-based, share one key issue. Their final output is either a set of pre-trained word embeddings based on the model hyper-parameters and the training corpus, or the trained network itself, that can be subsequently used to obtain embeddings for words. However, the relationship between embeddings and words (considered as types) is biunivocal. For each word type, one and only one embedding is available, and vice-versa. This is an inherent limitation to all such language models, and improvements to such paradigm have been the focus of a lot of research, especially in the last few years. In this case, the challenge is to incorporate context-specific information in the embeddings representation. Intuitively, such models should be able to generate an embedding for a specific word in a specific sequence of words, rather than in isolation. This is also crucial because most NLP downstream tasks

actually benefit from the understanding of both words and their contexts (Wang et al., 2019). One straightforward example is machine translation, where the order of words changes from language to language, and context-sensitive information is crucial to provide the best possible translation for a word.

Context-aware language models can be considered as a major breakthrough in the NLP field for several reasons. First, such models can take into account and thus encode in the representation of each word also all of its specific contexts. Typically, in fact, they accept as input entire sequences of tokens, and produce a contextualized hidden representation of each token. Second, they can be exploited as machine learning models to solve downstream tasks in NLP. The *Transformer* architecture in particular (Vaswani et al., 2017) has been extensively exploited and has obtained state-of-the-art results in many tasks, drastically outperforming most previously proposed machine learning models and paradigms (Devlin et al., 2019; Radford, 2018; Radford et al., 2019; Brown et al., 2020). Often, such architectures are in fact developed with downstream tasks in mind rather than language modelling itself. In this regard, it must be noted that such models have also allowed for the introduction of the pre-training and fine-tuning learning paradigm for NLP tasks, also known as *transfer learning*. In the case of transfer learning, a general language model trained on unsupervised language understanding tasks is further tuned to solve specific tasks. The idea is to retain the general language knowledge obtained during the pre-training and to actually transfer it over to the more specific tasks, by further training and tuning the weights and parameters of the network to address the more specific goal.

One of the first models of this kind proposed in the literature has been *ELMo* (Embeddings from Language Models) (Peters et al., 2018). Unlikely previously proposed models, word representations in ELMo are a function of the entire input sequence. The architecture is based on two Bidirectional LSTM (BiLSTM). Authors call the model BiLM. A BiLSTM is a sequence processing model that is composed of two LSTM layers, one that reads the sequence from the beginning to end, and the other one that reads it backwards. Each neuron in the layer outputs a representation that depends on all the previous neurons used to encode the sequence. After the input is passed trough the two LSTM layers, a vector representation for each element of the sequence is obtained, where each element is a function of all the previous and subsequent ones. BiLM employs two BiLSTM architectures to obtain two intermediate representations. Then, ELMo produces the final representation as the weighted sum of the original (and out-of-context) word vectors and the two intermediate representations. Authors report state-of-the-art performances on six supervised NLP tasks by using pre-trained ELMo models.

Arguably, one of the most important breakthroughs for the NLP community has been the introduction of the Transformer architecture (Vaswani et al., 2017),

a deep neural network for sequence transduction based entirely on the *attention* mechanism. Authors argue that the vast majority of state-of-the-art algorithms for sequence modelling and transduction are based on complex recurrent or convolutional neural networks and on the encoder-decoder architecture (Luong et al., 2015; Wu et al., 2016; Jozefowicz et al., 2016). Many such approaches include the attention mechanism in their architecture, in order to disregard the position and distance between dependencies in the input and output sequences (Bahdanau et al., 2015; Kim et al., 2017). The attention mechanism in fact is based on a function mapping a set of key-value pairs to a query, that can align vectors in the input and the output sequences by using a context vector (Bahdanau et al., 2015). In their work, Vaswani et al. (2017) propose a simple feed-forward encoder-decoder neural network that implements only a multi-head self attention mechanism to map input and output values. In practice, they propose the use of two feed-forward networks, one for the input sequence and one for the output sequence, that are linked via such attention. The input for both networks are embeddings for words in the sequence and an encoding of their position within the sequence. The output of the encoder network is passed trough several feed-forward and attention layers, and then used as input for the multi-head attention layer in the output network. The multi-head attention replaces recurrent and convolution layers in order to jointly learning different representations of the input and the output from different positions in the sequence. Figure 2.2 shows the Transformer architecture proposed by Vaswani et al. (2017).

Authors argue that their architecture poses the fundamental advantage of reducing the computational complexity in each layer of the network, and enabling the computation of long-range dependencies between words. This is because, while recurrent neural networks have to actually traverse the layer to identify long-range dependencies, the self-attention layer allows for a constant computational complexity when computing dependencies. Authors report state-of-the-art results on machine translation tasks.

While Vaswani et al. (2017) focused mostly on sequence-to-sequence models, the intuition behind the Transformer model and the multi-head self-attention mechanism inspired a huge number of researches focused on other aspects of language modelling. Radford (2018) introduced the pre-training and fine-tuning paradigm in NLP tasks. Specifically, authors propose GPT, an architecture based only on the decoder part of the Transformer (Liu et al., 2018), to solve a wide variety of NLP tasks. In GPT, the pre-training step consists in training the Transformer decoder with stochastic gradient descent on a standard language modelling objective. Pre-training is therefore unsupervised. Once the network has been pre-trained, the fine-tuning step consists of adding a linear output layer on top of the transformer, and update the weights of the whole network based on a supervised learning task. The input is always a sequence of tokens, that can be modelled depending on the task

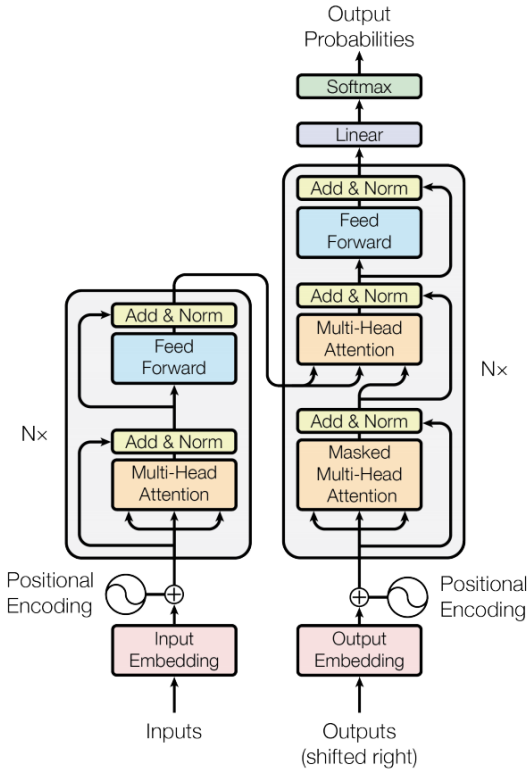


Figure 2.2: The Transformer architecture (Vaswani et al., 2017).

at hand (e.g. simple sequence classification, question answering, lexical entailment etc.). Moreover, depending on the task, multiple Transfomers can be exploited to encode the various inputs.

BERT (Bidirectional Encoder Representation for Transformers) (Devlin et al., 2019) takes the idea of pre-training and fine-tuning further. OpenAI GPT (Radford, 2018) uses a left-to-right architecture, and thus each token attends only to the previous tokens in the sequence. This means that the system may perform poorly when modelling sequence-level tasks. The authors of BERT propose a different pre-training strategy in order to obtain a bidirectional encoding of the whole sequence, based on the so called Masked Language Model and next-sentence prediction task. Specifically, during training tokens in the sequence are randomly substituted with a special [MASK] token. The training objective is then to learn to predict the correct token. This is used for the actual learning of the language model. As for the next-sentence prediction task, the training objective is, given a pair of sentences, to predict if they are adjacent or not. This is especially useful when applying the model to sentence-pair tasks. Authors propose two versions of the architecture, namely `bert-base` and `bert-large`, that differ in number of layers, parameters and resulting hidden representation. As for Radford (2018), during fine-tuning, an additional layer is included in the architecture, and pre-trained parameters are fine-tuned on

the supervised learning task. BERT is shown to obtain state-of-the-art results in both sentence-level and token-level tasks thanks to the bidirectional nature of its pre-training.

The introduction of BERT has sparked a lot of interest in the research community, given its effectiveness. A big chunk of research has focused on studying how the Transformer, and specifically BERT, works, in terms of syntactic and semantic information encoded in BERT components (i.e. layers, output embeddings, multi-head attention etc.) (Lin et al., 2019; Hewitt and Manning, 2019; Tenney et al., 2019; Ettinger, 2020; Mickus et al., 2019), pre-training and fine-tuning strategies (K et al., 2019; Liu et al., 2019; Raffel et al., 2020; Conneau and Lample, 2019; Kovaleva et al., 2019), and optimal number of parameters (Clark et al., 2019; Voita et al., 2019). An additional, and rather interesting aspect, concerns also techniques for compressing the BERT knowledge in smaller and less expensive models. This is achieved through several techniques, such as knowledge distillation (Hinton et al., 2015; Sanh et al., 2019), pruning (Guo et al., 2019), and quantization (Zadeh and Moshovos, 2020) among others. For example, DistilBERT (Sanh et al., 2019) is a widely popular model that uses knowledge distillation (Hinton et al., 2015) to obtain a model that has half the number of layers as the original BERT while retaining most of its performances. Typically, in a knowledge distillation framework, a smaller model, called the student, is trained to reproduce the behaviour and predictions of a bigger model, often named the teacher (Sanh et al., 2019). For a thorough review on the literature concerning the so-called BERTology, the interested reader can refer to Rogers et al. (2020).

In addition to BERT, several other Transformer models have been proposed in the literature. Three main directions can be broadly identified. First, as previously mentioned, several models based on distillation such as DistilBERT (Sanh et al., 2019) and ALBERT (Lan et al., 2020) have been proposed. These models are often focused on decreasing the computational cost of training and running such architectures, in order to allow for this kind of language modelling also on less performing systems. Second, several approaches have been proposed to modify some aspects of BERT in order to improve its performances while not increasing the computational cost. The most popular example in this case is RoBERTa (Liu et al., 2019). Authors argue that the original BERT model is undertrained, and demonstrate the effectiveness of a more robust pre-training on a numnber of taks. Finally, a whole line of research is dedicated to exploring the performance of increasingly bigger models, with orders of magnitude more parameters that the original BERT. In fact, while several researches argue that BERT and in general Transformer models are clearly overparametrized (Voita et al., 2019; Clark et al., 2019), many works have pointed out how increasing the size of Transformer models have brought also an increase in performance. Examples of such idea are Transformer XL (Yang et al., 2019), XLNet

(Dai et al., 2019), BigBird (Zaheer et al., 2020), and the iterations of the GPT architecture (Radford, 2018; Radford et al., 2019; Brown et al., 2020). Especially the latter architecture shows how, with a substantial increase in number of parameters, pre-trained Transformer models are able to achieve few-shot and even one-shot learning capabilities without fine-tuning (Brown et al., 2020). Apart from the fact that the computational cost for pre-training such models is an exclusive prerogative of large companies such as Google and OpenAI, it is interesting to notice how authors state that the relationship between computational cost (in terms of number of parameters in the network) and performances follows a power law that, even for the largest available model with 175 Billions of parameters, appears to not have reached a plateau just yet, and that the obtained results suggest that larger language models may prove crucial for developing general language systems (Brown et al., 2020).

As previously mentioned, all these modern architectures for language modelling have several advantages that should be stressed. First, they allow for modelling entire sequences of texts, while retaining information concerning the context for each element of the sequence and its context-aware meaning. Second, they have allowed for the pre-training and fine tuning paradigm to be applied to NLP taks. Third, they have shown state-of-the-art performances in most NLP tasks. The hidden representation of the whole sequence has proven to be rather reliable in solving many NLP problems when fine tuned in conjunction with a final sequence or token classification layer. Finally, while the main goal in developing transformer was to tackle downstream NLP tasks, they can be also used for feature extraction. In fact, hidden representation are available for each element of the input sequence, and can be used as context-aware word embeddings.

Embedding sentences and documents

The development of language modelling and word embedding models can be considered an invaluable milestone in the NLP literature. The advancements of the last few years in this regard have also led a considerable portion of the literature in the direction of providing the same encoding capabilities also for larger units of texts, such as sentences, paragraphs, and entire documents. In fact, the possibility of encoding, and thus comparing, larger portions of texts may prove invaluable for a variety of tasks, such as text clustering, information retrieval and extraction, and more generally for unsupervised techniques that rely on the semantics of entire sequences of texts. As for word embeddings, the end goal is to represent units of texts of varying length in an n-dimensional space.

The challenge of representing an entire sequence of words with a fixed-size continuous representation has been tackled from several different perspectives, that are often inspired by literature on word embeddings. Actually, earliest approaches were proposed in the information retrieval literature even before the ones regarding

words, with the already mentioned pioneering works of Harris (1954) and Salton et al. (1975). For most traditional approaches, documents are represented as bag-of-words. Given a pre-determined vocabulary containing n words, documents are represented as n-dimensional vectors in which each dimension refers to a word of the vocabulary being actually present or not in the document. In its simplest form, only the present option is reported. Other approaches proposed the use of raw frequencies. However, as it is true for word embeddings, raw frequencies have proven to be a rather unreliable measure also for document-wise vectors. Therefore, several term weighting techniques have been proposed when considering documents, including the widely popular tf-idf (term frequency-inverse document frequency) (Salton and Buckley, 1988).

Another widely popular technique for representing entire documents is LDA (Blei et al., 2003). Despite the fact that its main goal is to provide a technique for unsupervised topic discovery in documents, it can arguably be used as an embedding space for document corpora. LDA is a three-level hierarchical Bayesian model, in which documents are modelled as a distribution over a pre-determined number of topics, while topics are characterized by distribution over the words of the vocabulary. Thus, given n topics, the document can be represented as an n-dimensional vector where each dimension represents the probability of the n^{th} topic for the document.

Bag-of-words vectors with tf-idf weighting have gained wide popularity since their introduction. However, with the artificial intelligence revolution and the introduction of neural network language models (Bengio et al., 2003) such as word2vec (Mikolov et al., 2013a,b), also the processing of sentences and entire documents have shifted towards the creation of similar models that could encode documents as well. Already in Mikolov et al. (2013b) the intuition behind word2vec was applied also to n-grams embeddings, by extending the Skip-gram model to encode also bi-grams and tri-grams. Clearly, the method had limitations due to the inability to generalize to unseen phrases and to longer sequences.

The first attempt to generalize the word2vec algorithms to longer sequences is represented by *doc2vec* (Le and Mikolov, 2014). Two algorithms are proposed by the authors in order to incorporate the document information in the Skip-gram and CBOW algorithms, namely PV-DM (Paragraph Vector - Distributed Memory) and PV-DBOW (Paragraph Vector - Distributed Bag of Words). In PV-DM, as for the Skip-gram algorithm, the training goal is to predict a single word given the context. However, in addition to the vectors for words in the context window, a shared paragraph vector is learned during training with its respective contexts and target words. Further, instead of averaging the representation in the hidden layer, Le and Mikolov (2014) proposes to concatenate the paragraph and context vector, thus retaining word ordering. On the other hand, in PV-DBOW the training task is to predict a

single context word based only on the paragraph vector. In this case, word vectors are not learned along with the paragraph vector. A visual representation of PV-DM and PV-DBOW is presented in Figure 2.3

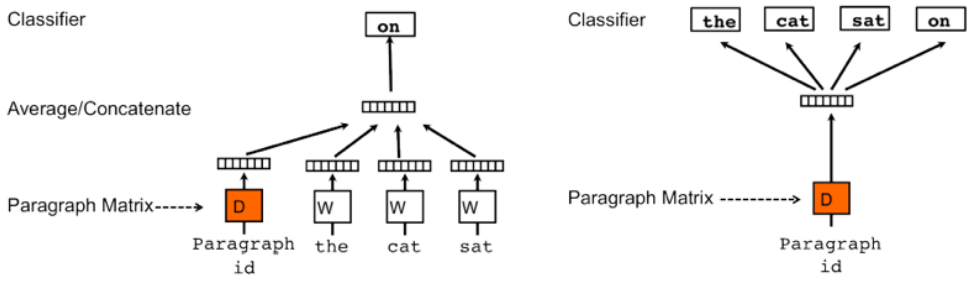


Figure 2.3: PV-DM (left) and PV-DBOW (rigth) (Le and Mikolov, 2014)

Several other methods and algorithms to adapt word2vec, and specifically Skip-gram, have been proposed in the literature, such as Skip-tought (Kiros et al., 2015), FastSent (Hill et al., 2016), and Quick-tought (Logeswaran and Lee, 2018) among others. In additon, Wu et al. (2018) propose the Word Mover Embedding technique, that implements Word Mover Distance (Kusner et al., 2015) to learn continuous representation of texts with varying length.

Finally, deep learning based models have been proposed, often achieving state-of-the-art performances in sentence representations and sentence similarity tasks. In this case, the direction of such models is to learn representation both from unlabelled data, as a language modelling task, and from labelled data to improve on the sentence-pair similarity. The intuition behind this approach has been first proposed in the context of machine translation, with works such as Cho et al. (2014); Sutskever et al. (2014), where sentence embeddings are learned from labelled parallel corpora for the machine translation task with an encoder-decoder architecture. The same intuition has been exploited also on a single language, for example on tasks such as learning paraphrases (Wieting et al., 2016), question answering (Das et al., 2016), and other Natural Language Inference tasks (Conneau et al., 2017; Nicosia and Moschitti, 2017). Transformers have also been employed, obtaining state-of-the-art results. As previously mentioned, the Transformer architecture can be used both for downstream tasks and to perform feature extraction in the form of word embeddings for each element in the input sequence. A representation of the whole sequence is typically available in the form of special tokens at the beginning or at the end of the sequence, that encodes information concerning the attention layers. However, as clearly stated in Devlin et al. (2019), sequence representation are not aimed at encoding semantically relevant information concerning the sequence, and thus are unsuitable for sentence-pair similarity tasks. Cer et al. (2018) proposed the Universal Sentence Encoder, which learns sentence representation on a supervised task for sentence similarity by either using the Transformer architecture or a Deep

Averaging Network model (Iyyer et al., 2015). GPT and its subsequent iterations (Radford, 2018; Radford et al., 2019; Brown et al., 2020) have also been studied on the task of sentence representation. Finally, Sentence-BERT (Reimers and Gurevych, 2019) is one of the most widely exploited architectures for learning sentence-wise representation. Sentence-BERT leverages Siamese BERT networks to obtain semantically relevant representations of sentences. Specifically, starting from a pre-trained Transformer model, it is fine-tuned on sentence-pair tasks with either a classification or regression objective function. As for the classification, given two sentences, with a label determining their similarity, the objective function must assign the correct label of similarity to the two sentence. As for the regression, the objective function is aimed at assigning the highest possible cosine similarity to the two resulting vector. In both cases, authors show that the best representation for sequences is obtained by averaging the word embeddings obtained by the BERT-based model. Thus, by fine-tuning the two BERT-based transformers and the final layer, authors are able to achieve state of the art performances in sentence similarity tasks. Authors propose to use the SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2018), and STS benchmark dataset (Cer et al., 2017) for effectively training the model. A python implementation [1] and several pre-trained Sentence-BERT models are available, both mono lingual and multi lingual (Reimers and Gurevych, 2020).

2.2 Fake News and Rumour Detection Techniques in the Literature

Aside from language modelling, and the use of pre-trained models to solve word and sentence similarity tasks, it is important to address also several questions concerning one of the most important aspects with respect to the present work, namely the fake news detection problem. One of the main goals of the present work is in fact that of exploiting language models to address the problem of fake news detection. As the literature concerning this specific issue is thriving in the last few years, it is important before going any further to address several key aspects of fake news and rumour detection and the approaches proposed to solve it.

Fake news and rumours have become widespread on social media in the last few years, due to the fact that social media are becoming more and more relevant as tools for news consumption, as shown by several case studies such as Newman et al. (2012). According to Zubiaga et al. (2018a), social media have become a critical publishing tool for journalists (Diakopoulos et al., 2012; Tolmie et al., 2017) and the main consumption method for citizens looking for the latest news (Hermida, 2010). Journalists may use social media to report on public opinions about breaking

[1] https://github.com/UKPLab/sentence-transformers

news stories, and even to discover potential new stories, whereas citizens may follow the development of breaking news and events through official channels (i.e. news outlets official accounts on social media platforms) or through posts of their own network (e.g. friends, family, public figures). Indeed, social networks have proved to be extremely useful especially during crisis situations, because of their inherent ability to spread breaking news much faster than traditional media (Vieweg, 2010). However, the absence of control and fact-checking over posts makes social media a fertile ground for the spread of unverified and/or false information (Zubiaga et al., 2018a), such as cases reported in Wang (2017) and Kang and Goldman (2016), that can in turn influence the public opinion on sensitive topics such as presidential elections (Allcott and Gentzkow, 2017).

Both social media platforms and the research community are very active on the topic of identifying potential fake claims and assessing their veracity. Fake news can take several different forms and shapes in the social media environment, such as rumours and clickbait articles, thus it is even more difficult to efficiently detect and contrast them, both manually and automatically. The research interest for fake news and rumour detection has in fact considerably grown in the last few years, as it is clear by the number of scientific publications on the topic.

Fake News and Rumours: terminology and definitions

In order to provide some insight on the issue of fake news, an analysis of the terminology may prove to be rather helpful. In the literature, different categorizations of fake news and rumours have been proposed, mostly depending on source and type of data used for analysis. Early studies in this field, especially from a computational perspective, are relatively recent. Therefore, the boundaries of the study matter are often not clearly defined. For this reason, we believe that it is fundamental to provide some insight into what kind of data can become the matter of analysis and how to define it.

Fake News. "Fake news" has become the de-facto expression for identifying general false information in mainstream media, especially for web-related content, mostly spreading during and after the 2016 U.S. Presidential Campaign.

However, research on fake news generally adopts a more restrictive definition. Following Allcott and Gentzkow (2017), a fake news is "a news article that is intentionally and verifiably false". Such definition hinges on two key aspects: *intent* and *verifiability*. Fake news are therefore news articles that are intentionally written to mislead or misinform readers, but can be verified as false by means of other sources. Several recent studies, such as Conroy et al. (2015), have adopted this definition. Authors also distinguish among different aspects of fake news, such as *serious fabrications*, *large scale hoaxes* and *humorous fakes* (Rubin et al., 2015).

Three key aspects can be identified concerning fake news: i) its form, as news article; ii) its deceptive intent, that can be either satirical or malicious; and iii) the verifiability of its content as completely or partially false.

Rumours. The term "rumours" on the other hand has been the most widely used and studied regarding the recent scientific literature. Rumours refer to information that has not been confirmed by official sources yet and is spread mostly by users on social media platforms. Earliest researches on rumours actually date back to the end of Word War II (Allport and Postman, 1946, 1947), but arguably the internet and especially social media platforms are the most fertile ground for the spread of such kind of unconfirmed or unverified information (Vosoughi et al., 2017).

Several interpretations concerning a formal definition of rumours have been provided in the literature. While some works identify rumours as simply circulating false information (Cai et al., 2014), making them more akin to fake news, most definitions consider other aspects, Di Fonzo and Bordia (2007) identify rumours as "unverified and instrumentally relevant information statements in circulation". Zubiaga et al. (2015) defines a rumour more specifically as a "circulating story of questionable veracity, which is apparently credible but hard to verify, and produces sufficient skepticism and/or anxiety". For this definition a rumour has to produce an impactful reaction on its audience. Arguably, both these definitions hinge on the "unverified" characteristic of the information being shared. This unverified information could be true, partly true, entirely false or remain unverified (Zubiaga et al., 2018a). From a more practical standpoint, Zubiaga et al. (2018a) has also interestingly split rumours into the two categories of *long standing rumours*, that represent unverified information circulating for long periods of time (e.g. conspiracy theories), and *breaking news rumours*, that often appear in connection with breaking news stories, and could either be the product of unintentional misinformation or intentional deception.

Overview of the most common approaches

The most common approaches for fake news and rumour detection tend to treat the task as a classification problem: the goal is to assign labels such as *rumour* or *non rumour*, *true* or *false* to a particular piece of text. In most of the cases, researchers have employed machine learning and deep learning approaches, achieving promising results. Alternatively, some researchers have applied other approaches based, for instance, on data mining techniques, such as time series analysis, and have exploited external resources (e.g. knowledge bases), to predict either the class of documents or events, or to asses their credibility. In this regard, it is important to point out also that among alternative approaches, *fact-checking* has played an important role especially in the last few years.

In addition, it is important to notice that the interest from the research community has also sparked several competitions with the goal of evaluating approaches to determining the veracity of content in social media (Derczynski et al., 2017; Gorrell et al., 2019; Hanselowski et al., 2018) and concerning fact-checking related tasks (Elsayed et al., 2019; Barrón-Cedeño et al., 2020).

Machine Learning models for classification

An important distinction among approaches should be made based on what features are considered relevant to detect fake news and rumours. In Shu et al. (2017) a distinction is made between *content-based* and *context-based* approaches. On the one hand, content-based approaches rely on *content features*, which refer to information that can be directly extracted from text, such as linguistic features. On the other hand, context-based approaches are more varied, and generally rely on surrounding information, such as user's characteristics, social network propagation features and reactions of other users to the news or post.

Content features are generally extracted directly from the text itself. Linguistic cues of deception have been widely studied in NLP (Briscoe et al., 2014; Pérez-Rosas and Mihalcea, 2015; Zhang et al., 2012). Such cues are for example the use of self reference, swear words and negative words. They have proven their reliability in tasks where the intention of the author must be established, such as for example sentiment analysis. There is a rather large body of work in which such features have been used, in isolation or in conjunction with and have been proposed as a viable option also for fake news and rumour detection (Castillo et al., 2011; Gupta et al., 2014; Hamidian and Diab, 2015; Ma et al., 2016; Qin et al., 2016; Rubin et al., 2016; Volkova et al., 2017; Zhang et al., 2012). Syntactic and lexical features such as the presence and frequency of certain words and patterns have been used as means to identify fake news based on the linguistic properties of texts (Qazvinian et al., 2011; Zubiaga et al., 2016; Chua and Banerjee, 2016; Hardalov et al., 2016; Feng and Hirst, 2013; Potthast et al., 2016). Finally, semantic information such as topics and word embeddings have proven to be useful in numerous contexts throughout the NLP research field, and have been successfully implemented also for fake news and rumours detection, especially in machine learning and deep learning approaches (Jin et al., 2017; Ma et al., 2016; Ruchansky et al., 2017; Zubiaga et al., 2016).

As for context-based features, they are often exploited in approaches that do not directly rely on NLP methods. Relevant features are considered those that surround the actual social media post or fake news. In particular, the most used context features concern the analysis of users, sources of the rumour or news, propagation structures of the information on social media, and reaction of other users with respect to the news. The analysis of users focuses mostly on their activity on social media (e.g. number of interactions and posts) (Kwon et al., 2013; Zubiaga et al.,

2016) and their profile (e.g. age of the account, the description, and URL linking to external resources) (Castillo et al., 2011; Chang et al., 2016; Liu et al., 2015; Ma et al., 2015; Wu et al., 2015; Yang et al., 2012; Zubiaga et al., 2016). The analysis of the network in which fake news and rumours are spread mostly focuses on propagation structures, diffusion patterns, properties of the sub-graph in which the news is spread, and propagation times (Castillo et al., 2011; Hamidian and Diab, 2015; Ma et al., 2015; Yang et al., 2012; Wang and Terano, 2015; Wu et al., 2015). Finally, also the stance regarding specific posts have been exploited in the literature, albeit scarcely, as features to model fake news detection algorithms (Zubiaga et al., 2016; Jin et al., 2016; Tacchini et al., 2017).

According to Shu et al. (2017) many fake news detection approaches mostly rely on content-based features. On the contrary, given the more social nature of rumours, approaches focusing on the task of identifying them in streams of social media posts are more prone to exploit both content-based and context-based features for the analysis.

Given the fact that the problem of fake news and rumour detection and its subtasks, such as for example stance detection, have been mostly considered a standard classification problem, most of the approaches focused on the implementation of machine learning strategies to solve it. Earliest approaches focused on the application of traditional machine learning algorithms, such as Support Vector Machines (SVM) (Afroz et al., 2012; Briscoe et al., 2014; Pérez-Rosas and Mihalcea, 2015; Rubin et al., 2016; Zhang et al., 2012; Qin et al., 2016; Yang et al., 2012; Wu et al., 2015; Horne and Adali, 2017), Decision Tree or Random Forest (Aker et al., 2017; Castillo et al., 2011; Giasemidis et al., 2016; Zhao et al., 2015), Conditional Random Field (CRF) (Zubiaga et al., 2016; Zubiaga et al., 2017) and Hidden Markov Models (HMM) (Vosoughi, 2015; Vosoughi et al., 2017).

On the contrary, many modern models rely on deep neural networks architectures. Deep learning frameworks have a main advantage over traditional machine learning approaches. Indeed, traditional machine learning representations are based on manually crafted features. The feature extraction task is time-consuming and may result in biased features (Ma et al., 2016). This is a critical issue for tasks such as fake news and rumour detection, where the identification of relevant features for the analysis may pose an even greater challenge. On the other hand, deep learning frameworks can learn hidden representations from simpler inputs both in context and content variations (Ma et al., 2016). The problem is therefore shifted from modeling relevant input features to modeling the network itself in a way that enables the task to be solved efficiently. In the fake news and rumour domain, both Recurrent Neural Networks and Convolutional Neural Networks have been proposed, as well as ensemble hybrid approaches that employs both, and have obtained very competitive performances, consistently outperforming standard ma-

chine learning models (Ma et al., 2016; Ruchansky et al., 2017; Chen et al., 2017; Yu et al., 2017; Volkova et al., 2017; Wang, 2017; Zubiaga et al., 2018b; Kochkina et al., 2018; Ajao et al., 2018; Song et al., 2018).

Finally, it is important to notice that since its introduction, the Transformer architecture, equipped with its transfer learning capabilities, has been applied also to the fake news domain, obtaining state-of-the-art performances (Slovikovskaya and Attardi, 2020; Qazi et al., 2020).

Fact-checking

In this context, it is also worth mentioning that an important part of the literature concerning fake news and rumours revolved around the task of computational-oriented fact-checking (Shu et al., 2017). Although the task of fact-checking can be considered as slightly different from fake news and rumour detection from a technical standpoint, it is nonetheless definitely akin to them. The main effort is addressed to perform automatically fact-checking. A number of strategies have been proposed to this aim. Magdy and Wanas (2010) has automated the process of web-based fact-checking, by comparing facts extracted from a given document against facts extracted from URLs related to such documents. Wu et al. (2014) has introduced the task of automated fact-checking and presented a series of algorithms for solving the task by automatically designing queries aimed at checking whether the statement is true or false. The most widely used technique is however the exploitation of knowledge graphs. Such graphs have been employed in a number of studies (Ciampaglia et al., 2015; Shi and Weninger, 2016). More specifically, the use of Wikipedia *infoboxes* to generate a knowledge graph has been proposed in Ciampaglia et al. (2015). Here, the authors have defined a measure of semantic proximity by using a transitive closure algorithm in order to check claims against the knowledge graph. In Shi and Weninger (2016) the problem has been tackled as a link prediction task in a knowledge graph. Each statement corresponds to a path in the graph: the existence of meta-paths is exploited with the aim of reducing the search space with respect to the statement's path. As previously mentioned, several competitions have been aimed at assessing approaches to computational fact-checking, such as Elsayed et al. (2019); Barrón-Cedeño et al. (2020). Many of the best performing systems in these competitions have shown that approaches based on deep learning and the Transformer architecture are rather effective to retrieve verified claims for a given unverified piece of news or text.

Social media based Data collection in the literature

One of the key issues when tackling fake news and rumour detection from a computational perspective is clearly the collection of data. Researchers have to face a series

of issues. First, they have to manage different types of false information in the context of web and social media platforms. For example, rumours on social networks, fake news articles on malicious websites, fake reviews, etc. Second, the amount of false information is a small fraction of online content produced every day, even if we restrict our focus on news articles and posts discussing breaking news. Third, social media companies have nowadays strict policies for what concerns the analysis of data produced by their users. This is especially true after the Facebook and Cambridge Analytica data scandal surfaced in the first months of 2018. Finally, given the different types of misinformation, several different tasks have been proposed in the literature, such as fake news detection, clickbait detection, rumour detection, and rumour veracity classification. For each task, different means of collecting and annotating data may be necessary. For these reasons, a few benchmark datasets and data repositories are today publicly available.

As for fake news, the main source of content is clearly malicious websites, specifically created to spread misinformation. Their articles are often later shared on social media platforms by authors, malicious users working with them or social media bots, and inattentive users who do not bother to check the source of the article before sharing it. Therefore, given some knowledge about certain fake news, they could be directly gathered by crawling social media. Certain fake news websites are built to resemble proper news outlets, by mimicking both the visual aspect and the domain name. For this reason, such websites can be used to harvest articles, which have a high probability of being false. However, arguably inferring the veracity of a piece of news solely based on its source could be misleading. Moreover, fake news may also be found on verified sources. This could happen for example by mistake, or for the rush of publishing breaking news without properly checking sources beforehand. Thus, it is clear that a proper annotation of data, to be conducted by professionals with knowledge on the matter and access to many different sources, is strongly advisable. Clearly, as pointed out by both Rubin et al. (2015) and Shu et al. (2017), a key aspect that should be considered when gathering reliable data for fake news detection is to clearly assess the veracity of each element of the dataset. Expert-oriented and crowdsourcing-oriented fact-checking can be exploited to reliably annotate datasets of fake news.

As for rumours, they are often studied directly on social media, as the case is often that they are born there, rather than on external sources. Social media platforms are often used to share information as quickly as possible between users. This may result in sharing unverified information that, in turn, may spread and generate a rumour. The literature on how datasets should be collected for rumours detection and analysis mostly focuses on two main strategies, namely *top-down* and *bottom-up* collection strategies (Zubiaga et al., 2018a). Top-down strategy requires some form of a-priori knowledge about target rumours. In particular, rumours are usu-

ally collected, after they spread on social media, by searching for a specific set of *keywords* and *tags* that describe the rumour. The proposed strategy is quite straightforward to implement and thus has been employed in several researches related to rumour detection and verification (Castillo et al., 2011; Ma et al., 2015; Qazvinian et al., 2011; Vosoughi et al., 2017; Zhao et al., 2015). In this context rumour debunking websites are often used as source to identify the most interesting rumours and to extract reliable keywords for retrieving posts about those rumours on social media (Chua and Banerjee, 2016; Jin et al., 2017; Kwon et al., 2013; Liu et al., 2015; Ma et al., 2016; Qazvinian et al., 2011). Clearly, this approach has several drawbacks. First, rumours must be known a-priori, at least in some form. Second, social media limitations make it difficult to retrieve huge collections of data on specific topics. The top-down approach may be nonetheless efficient, especially for long-standing rumours. On the other hand, a bottom-up approach is specifically aimed at collecting potential rumours in breaking news. This collection strategy has been proposed in Derczynski et al. (2017); Giasemidis et al. (2016); Zubiaga et al. (2016); Zubiaga et al. (2016). The main idea is to gather as many social media posts as possible during a certain time window, and then let expert human annotators label them with various levels of granularity (e.g. rumour/non rumour, true/false, etc.). Despite being clearly advantaged by the fact that it is not relevant to know rumours a-priori, bottom-up is more costly in terms of human annotators and resources, and may result in a limited number of rumours collected during the specific period of time (Zubiaga et al., 2018a).

Publicly available datasets

As far as collected and labelled datasets are concerned, not many resources are publicly available. This may depend from several factors. First, it is clearly not easy to identify relevant data and propose effective strategies to collect them. Second, no agreed-upon definition of fake news and rumour is available. This may pose a challenge when labelling the data. Finally, as data are found especially in social media, and such platforms are restrictive when giving access to their data, both for its strategic and economical value and for protecting the privacy of users, the collection process become a rather relevant challenge.

For what concerns fake news, Shu et al. (2017) states that an agreed upon benchmark dataset for fake news detection has not been produced yet. However, several publicly available resources are worth mentioning. Many authors have focused on the creation of datasets containing statements from social media, for example made by politicians or public figures, labeled with information about their veracity for performing fact-checking and fake news detection (Vlachos and Riedel, 2014; Wang, 2017). Similar synthetically produced datasets are available as well (Thorne et al., 2018). Other authors have focused on hyperpartisan (Potthast et al., 2016)

and pseudo-scientific (Tacchini et al., 2017) publishers on Facebook. As for Twitter, a large scale dataset labelled for credibility judgments has been collected by Mitra and Gilbert (2015). Proper news articles were targeted by Ferreira and Vlachos (2016). Their proposed dataset contains a set of rumored claims and related news articles, annotated for their veracity judgments. The dataset proposed in Ferreira and Vlachos (2016) has been exploited also for the Fake News Challenge Stage 1 (FNC-1) task on stance detection. Finally, Shu et al. (2018) has provided the most complete dataset of statements in terms of related information. Authors have built a system for fake news detection, and provided a dataset containing information about the content (textual and visual), information about social context (i.e. users, network information, etc.), and characteristics of the spread evolution.

Concerning rumours, a more extensive effort has been undertaken in order to gather relevant data, especially in the context of the PHEME project (Derczynski and Bontcheva, 2014). The most relevant dataset, that can be considered as a benchmark for different possible evaluation purposes, has been collected by Zubiaga et al. (2016). The dataset includes tweets related to 9 different rumours collected from 2014 to 2015. A bottom-up strategy was followed to gather data. Subsequently, tweets with a high retweet count were annotated on different levels (rumour/non rumour, true/false/unverified etc.) by a group of journalists. Moreover, tweets were grouped by events and *stories* within each event. Each story has information about the *resolving tweet*, if present. The resolving tweet is considered as the one that, for the annotator, was decisive for establishing the veracity of the rumour (Zubiaga et al., 2016). In addition, responses to tweets were collected and annotated for stance with respect to the source tweet. The dataset has been used both in research (Zubiaga et al., 2016) and for the RumourEval task in the SemEval 2017 evaluation campaign (Derczynski et al., 2017). Finally, the previously mentioned CREDBANK (Mitra and Gilbert, 2015) may be a useful resource, especially concerning the veracity classification task of rumours on social media.

Potential shortcomings of current approaches

It is clear that, given the fact that fake news detection, rumour detection, and computational fact-checking, are relatively new topics for the research communities of Data Mining and NLP, several major issues and potential shortcomings of current approaches can be pointed out.

First, the lack of widely accepted benchmark datasets may hinder the ability of clearly assessing the quality of proposed models and strategies to detect and verify fake news and rumours. In fact, available resources may not be sufficient for i) gaining novel insight on relevant properties of fake news and rumours and ii) building models able to properly operate in a real world scenario. On the one hand, the production of large scale datasets could enable analysis on a level more similar to real

scenarios, and allow to better identify telltale signs of fake news and rumours that can help in generalizing approaches and models. On the other hand, the distribution of benchmark datasets may help researchers in better assessing and evaluating their models against gold standard data.

Second, a clear trend in favor of supervised classification approaches to fake news and rumours detection is visible in the literature. Deep learning techniques have obtained state-of-the-art results, and most recent approaches are focused on exploiting such frameworks to some extent. As for the feature engineering, there is still vast room of improvement in terms of generalization capabilities of the features. In fact, given the absence of benchmark dataset, many researches focused on modelling the problem on more limited data, that may not generalize well and be suitable in real-case scenarios. In addition to this, not many unsupervised or semi-supervised approaches have been proposed, that could arguably allow for a better generalization on large-scale datasets and a wider array of topics, and a better understanding of the problem and of its key characteristics. Modeling the problem from a different perspective may allow to overcome the limitations of the proposed classification approaches, namely the need for labeled training data and potential lack of generalization capabilities in a real world scenario.

2.3 Summary

In this Chapter, a review of the literature concerning the key topics discussed in this thesis is provided.

Section 2.1 provides basic concepts of textual similarity, both for words and entire sequences of text. The most prominent approaches are thoroughly discussed, with a specific focus on the ones exploited in the present work. Particular focus is also given to models that can be pre-trained in order to provide out-of-the-box capabilities of representing words and sequences as distributed vectors in an n-dimensional space. Moreover, state-of-the-art models are discussed.

In Section 2.2 a review of the literature concerning fake news, rumours fact-checking and related topics is proposed. Specific topics of interest are state-of-the-art models, available datasets and data collection strategies.

Chapter 3

Case study: Profiling city areas with news articles

The problem of modelling the semantics of words, and understanding how their meanings relate to each other has been at the forefront of NLP research since its earliest days. The distributional hypothesis, proposed in Harris (1954) and Firth (1957), was the first staple of future research. For the distributional hypothesis, *linguistic items with similar distributions have similar meanings*. Earliest computational approaches to such hypothesis were aimed at building representations for the meaning of words by exploiting co-occurrences analysis in corpora. More recently, machine and deep learning have opened new paths to compute the meaning of words as *word embeddings* (Turian et al., 2010). Word embeddings are a fixed-size distributed representation of words (or *sub-words*) in a multi-dimensional real space (Bengio et al., 2003; Mikolov et al., 2013a,b; Bojanowski et al., 2017; Joulin et al., 2017).

In this Chapter, a case study on the use of pre-trained word embeddings for categorization is presented, in order to perform the profiling of city areas by means of textual information contained in news articles. In the approach, word embeddings are implemented for categorizing news articles in several macro-areas of interest based on their tags. Such categories are then exploited and evaluated by training a machine learning classifier to label novel articles into these macro-categories.

The reason for the specific case study is driven by two main factors. On the one hand, the possibility of evaluating the quality of word embeddings, and especially of pre-trained models, in a rather simple yet useful context. On the other hand, its purpose is to assess how NLP techniques such as text classification can improve the quality and performances of other systems and frameworks not directly aimed at solving language-related tasks. In this case, it proves to be an added value in the creation of frameworks with the goal of improving the life of residents in the city by means of technology.

As cities all around the world grow both in size and population, it is becoming

increasingly important for local governments to access tools and frameworks for profiling city areas in terms of commercial and social activities, citizens' behaviour and issues. Such profiling may in fact be useful for supporting the decision making process of local administrations and security officers, as well as for citizens in a number of use cases, ranging from the improvement of services available to the community to the effective management of crisis situations. Having a clear and updated snapshot of what actually happens in a given area or neighbourhood of a city at a given time can improve the quality of services offered to citizens, thus in turn increasing their quality of life, and help in handling problems such as crime and weather threats in real time.

Profiling city areas is thus an important aspect especially in the context of *smart cities* (Silva et al., 2018; Giatsoglou et al., 2016). The concept of smart city generally hinges on the idea that a smart city should be able to provide the best quality of life and services to its citizens through the smart use of technologies. The last few years have seen a growth in the interest on smart cities from several different research communities, as different but interconnected domains play an important role in this context, such as for example the environmental, economical, social and transportation/mobility ones (Trasarti et al., 2011; Sakaki et al., 2013; Anastasi et al., 2013; D'Andrea et al., 2015; Yang et al., 2017).

The underlying idea behind profiling is that of making generalizations about items of interest based on characteristics and patterns found in the data. This enables to construct profiles with such information and to exploit them for the identification of similar items and categorization of new ones. For what concerns profiling of smart cities, several works have been proposed focused on specific aspects of the city life. For example, Giatsoglou et al. (2016) proposed CityPulse, a web platform that exploits large-scale data analysis in order to support decision making for both citizens and administrations. Analyzed data come from various online social sources, such as geo-located social media posts (e.g. tweets, check-ins, photos). In D'Andrea et al. (2018), a framework was proposed to allow for the collection of available geo-located data (about Points of Interest, posts, traffic information, etc.) from online web services and sites related to a specific city. By means of the framework, users can build a virtual grid over a specific portion of the territory covered by a city. Each cell of the grid represents an area, which is characterised by the information extracted from different online sources. For example, areas of the city can be grouped together and classified by the different Points of Interest (POIs) that are included in them.

In this specific context, NLP techniques may provide an invaluable advantage in that they allow modelling unstructured data, such as texts coming from various sources, including newspapers and social media, to identify critical aspects of the city life, considering both events that happen in the city and the perspective of cit-

izens on such events, and more generally their perception of the city they live in. This in turn may lead local governments to make informed decision on how to act in specific neighborhoods or city areas to improve the quality of life for their residents.

3.1 Methodology

The present work is based on the framework proposed in D'Andrea et al. (2018). The goal is to exploit the framework for profiling city areas over time by using data and meta-data extracted from articles of online newspapers.

The reason for choosing online newspaper is straightforward. Generally, web-based content is particularly useful when considering real-time applications that aim to build models able to evolve as new data, and therefore potentially new categories for it, are generated. In this regard, an incredibly valuable source of information is clearly online newspapers. In fact, online news is currently the most popular channel for Internet users to read news. The term *Web News Mining* has been recently coined (Iglesias et al., 2016) to identify the huge amount of valuable information that can be continuously mined from online news. Usually, several kinds of data can be extracted from the articles published on online newspapers. One of the main sources of information is clearly the text, including the title and the body of the piece of news. In addition to the actual text of the article and the date of publication, often images, videos, tags describing the categories of the news, comments written by the readers, information regarding the geo-location of the news, and other meta-data are included. This type of information can be extremely valuable for profiling tasks.

The end goal is to provide maps describing the different city areas in terms of specific events and issues, such as criminality, urban decay, traffic and accidents, and immigration problems. Such maps can refer to specific periods of time and specific areas of the city, thus providing the current situation of the city and the past and future ones, in order to enable comparison and evaluate the changes that happen in the city. To these ends, the *location* and the list of *tags* associated with news articles for the specific city are exploited. This allows, on the one side, to localize each news in a specific part, or cell, of the city, and on the other side to describe each cell by exploiting several macro-categories of news based on tags. This description is constantly updated as new pieces of news are collected and analyzed within the framework.

Framework description

A brief description of the framework is provided in the following in order to better understand how the data are obtained, analyzed and exploited to achieve the end goal.

The framework consists of four main modules, described below:

Data Retrieval – The data retrieval module is used for data collection from the web. Web sources are exploited either trough available APIs or via web scraping. The framework can handle streams of data, as for example is the case for news sources that are updated almost continuously.

Data Preparation – In the data preparation module, several operations are applied to the collected data for enabling further analysis. Specifically, the data are pre-processed, aggregated and filtered according to how the areas of the city are defined. Specifically, once the boundaries of the city are defined, a simple grid of squared cells is superimposed on the map. Cell size is defined by the user. For each cell, the raw data are used to extract relevant descriptive features of the city area.

Data Mining – The data mining module is used to perform analysis on the raw or aggregated data. Analysis includes several machine learning algorithms for classification, regression, clustering and so on.

Result Visualization – The result visualization module is finally used to visualize the raw or aggregated data, and the results obtained by the Data Mining module in terms of maps, graphs, charts and statistics.

Online newspapers as a data source

As previously mentioned, online newspapers are a rather interesting data source for text mining. They have in fact attracted a lot of attention from the research community in the last few years, due to several reasons. First, they are one of the most prominent information sources for citizens, because they provide an easily-accessible and always-up-to-date source of information, both for world news and more locally focused events. Second, from a research standpoint, they prove to be rather useful because both the data, i.e. the news itself containing text and images, and the metadata associated with it are interesting. Metadata typically contain geo-localization of the news, and tags associated with it. A dataset containing around 100,000 news articles and their metadata was for example proposed in Ramisa et al. (2018). Data and metadata from news articles could be exploited in a wide variety of data mining and text mining tasks, such as news categorization and localization of criminal activities (Ramisa et al., 2018; Po and Rollo, 2018; Lin et al., 2018).

For the present work, as the primary goal is to profile different areas within a specific city, the data source that was integrated into the framework was the Today

websites.[1] The group of websites is owned by the CityNews company. The main advantages of such websites is that they offer online news for 48 different Italian cities. Each website refers to a particular city. For example, `pisatoday.it` reports news for the city of Pisa, while `romatoday.it` contains news from the metropolitan city of Rome. The service is free and it is continuously updated. For our goal, two kinds of metadata available through the Today websites were extracted, namely the *location* and the list of *tags*. Given the location provided in plain text (e.g. "Via Appia Nord"), by means of the Geocoding API[2] provided by Google Maps, the exact geographic coordinates of the location can be extracted. As for the tags, they are provided with the articles themselves. Tags are words (or multi-words) that are used for describing a piece of news. For example, an article describing an attempted theft at a local drug store that was stopped by the police may contain tags such as "Theft", "Robbery", "Pursuit", and "Arrest". Tags are specified by the author of the article, and are not limited to any pre-determined list. Nonetheless, they are usually common words. Each article may contain zero or more tags.

Macro-categorization of articles based on tags

The key aspect pertaining to the present research is the identification of macro-categories to label each piece of news. In fact, as tags are defined arbitrarily by the author of the article, this can result in a huge amount of different tags within the same category of articles, city areas and online newspapers. In order to identify a set of macro-categories that can broadly specify the various topics each article is about, word embeddings are used in conjunction with clustering algorithms. This enables the identification of groups of tags associated with similar topics discussed in the articles.

The proposed approach stems from the fact that word embeddings have become the de-facto standard for representing the meaning of words in most NLP tasks. Albeit the evaluation of the quality of word embeddings depends on several interconnected factors, such as word relatedness, coherence and downstream performance, and that the evaluation should be contextualized on the task at hand (Schnabel et al., 2015), it can be argued that word embeddings have proven their reliability in encoding the semantics of words in a machine-readable format (Mikolov et al., 2013a; Baroni et al., 2014). Specifically, words are represented as real-valued vectors in a multi-dimensional real space. As the models used for building such vectors typically learn from the distribution of words and their contexts in corpora, it is expected that, in the resulting space, vectors of words that are typically semantically similar are closer to each other than vectors of words that bear different meanings. Prox-

[1] http://www.today.it/

[2] https://developers.google.com/maps/documentation/geocoding/start

imity among word vectors is typically computed in terms of *cosine* similarity, as it allows disregarding the magnitude of the vector and only considering its direction in the space.

In the present context, the selected word embeddings were generated via the neural approach popularized by Mikolov et al. (2013a,b). In such approach, neural networks are used to learn the embedding of words from unlabelled corpora of plain text. Specifically, authors propose two algorithms, namely Continuous Bag-of-Words (CBOW) and Skip-gram for learning the embeddings from unlabelled data in different fashion. Typically, learned models and representations are stored in the form of pre-trained word embeddings, that are often made available for download and usage. The approach proposed in Mikolov et al. (2013a,b) has one key disadvantage: out-of-vocabulary words, i.e. words that did not appear in the training corpus, do not have a representation in the pre-trained model. For this reason, several similar approaches have been proposed, including fastText (Bojanowski et al., 2017; Joulin et al., 2017). In fastText, sub-word information is taken into account as well. This has the effect of being able to model (i.e. obtain a word vector) also out-of-vocabulary words.

To obtain embeddings for article tags, a pre-trained fastText model was chosen. FastText provides pre-trained model for 157 languages (Grave et al., 2018). The model was originally trained on Common Crawl and Wikipedia texts. The original training hyper-parameters reported in Grave et al. (2018) were the following: CBOW algorithm with position-weights, vector dimension set to 300, character n-grams length of 5 and window-size length of 5.

Once the vectors for each tag are obtained, an *agglomerative hierarchical clustering* algorithm (Witten et al., 2016) is applied to the data with cosine as distance metrics, in order to find groups of tags with similar meaning. The result, obtained by cutting the resulting dendrogram at a specific height, is in fact a set of clusters based on the semantics of the tags obtained by the news articles. Each of the clusters identifies a macro-category. Macro-categories are then appropriately labelled by manually checking the tags that are contained in the specific cluster.

Text Categorization Model

Finally, in order to asses the quality of the macro-categories, and to provide the framework with a trained model able to assign a macro-category to each new article, a text categorization model is proposed.

A basic approach to macro-category assignment would be to simply select the cluster in which the majority of new articles' tags are projected. However, this approach may pose several problems. First, tags may not have been produced for the specific news article. Second, as tags are arbitrarily assigned by the author of the article, new words that are not mapped in the clustering could be used, thus intro-

ducing the need for assessing the similarity of new tags with entire clusters. Albeit simple to implement, such an approach would be rather hard to evaluate and to incorporate in most use cases. Finally, in the perspective of the framework, if other sources such as different online news websites are added, such sources may not adopt a tagging system for their articles.

In order to address such issues, a viable approach would be to exploit tags and the resulting macro-categories in a semi-supervised way. Specifically, a *text categorization model* is proposed that is able to assign one of the macro-categories to a previously unseen and potentially non-tagged news article. To this end, a multi-class SVM classifier is adopted, that uses bag-of-words representations for texts (D'Andrea et al., 2019). Specifically, we evaluated the performances, in terms of obtained results and computational cost, of two bag-of-words representations for news articles, namely a *Complete* one, consisting of the title and the entire text of the article, and a *Partial* one, consisting of only the title and the summary of the article, to ensure as little sparsity as possible in the representation. In fact, one of the main drawbacks of the bag-of-words representation is that the feature space grows linearly with the vocabulary. By keeping only the title and the summary of each article, it is possible to balance the size of the vocabulary, and thus the computational cost, with the quality and quantity of information, as most of the relevant notions in a news article are usually contained in the title and summary.

For the training of the text categorization model, a subset of the articles were used. Articles were labelled based on the macro-category which most of its tags are associated with.

3.2 Experiments and obtained results

In order to validate the approach, several experiments have been performed. The outcome of the experiments can help evaluating various aspects of the proposed method. First, it is possible to evaluate the quality of representations for pre-trained word embeddings, and the effectiveness of hierarchical clustering on such embeddings to categorize words, in this case news article tags, based on their semantic similarity and relatedness, and obtaining insights into what are the most interesting and broad categories of news reports. Second, we can evaluate the quality of a machine learning model based on labels (i.e. the macro-categories) that are assigned to training data in a semi-automatic fashion. Third, from the framework, and thus from a more application-based perspective, it is possible to evaluate the effectiveness of the profiling of neighborhoods and areas of a city based on news articles.

For the experiments, the Metropolitan city of Rome was selected as the subject of the analysis. Specifically, the main urban area of Rome was considered, i.e. inside and near its Ring Road ("Grande Raccordo Anulare").

Dataset

In order to obtain news articles for the city of Rome, the RomaToday[3] website was added as a source to the city profiling framework. The website provides news exclusively regarding the Italian capital. Articles from 2014 to 2018 were collected for the analysis, in order to allow for time variability and quantity of data. As no API is available for the website, articles and their metadata were collected by means of web scraping. Table 3.1 presents an article with its English translation and relative metadata collected from the website.

Type of data	**Value**	**English Translation**
Link	http://parioli.romatoday.it /salario/via-di-priscilla-chiusa-bus-deviati.html	
Time Stamp	09:04:2019-09:47	
Title	Roma sprofonda, chiusa via di Priscilla: quartiere in tilt	Rome collapses, closed via di Priscilla: neighborhood in chaos
List of Tags	Deviazioni Bus, Strade Chiuse	Bus deviations, closed roads
Location	Roma, Via di Priscilla	
Summary	La strada del Salario interdetta al traffico tra piazza Vescovio e via Monte delle Gioie: accertamenti tecnici in corso sull'avvallamento	The Salario road is closed to traffic between piazza Vescovio and via Monte delle Gioie: expert inspections in progress on the dip
Text of the news	Via di Priscilla chiusa a causa di un avvallamento. La strada del quartiere Salario è interdetta al traffico tra piazza Vescovio e via Monte delle Gioie...	Via di Priscilla closed to the traffic due to a dip. The street of the Salario district is closed to traffic between Piazza Vescovio and Via Monte delle Gioie...

Table 3.1: Data sample. A news article concerning roads and traffic.

Approximately 17,000 news articles were collected for the selected time span. Table 3.2 shows a summary of the extracted data. It is important to notice that more than 3,500 articles do not have any tags. This serves as a further proof that a text categorization model is actually crucial for the task at hand. Of the whole dataset, 11,675 articles from 2014 to 2017 were used as the training set for the text categorization model, while 2,791 articles produced in 2018 were left for testing. As for the tags, it is important to notice that the number of distinct tags identified is $\approx$2,900. Therefore, a subset of them was selected by considering their frequency in the dataset. By using a minimum frequency threshold of 50, 135 distinct tags were obtained. Finally, a further manual evaluation of the obtained tags was performed,

[3]www.romatoday.it

in order to remove the most common and uninteresting words for the analysis, such as for example "Roma", "via" (*road*), and so on. After the manual evaluation, 86 tags in total were considered as relevant. As for the distribution of locations, which is the other key metadata for the present analysis, Figure 3.1 shows a snapshot of the distribution of news in the city, visualized within the framework. The particular distribution refers only to data from 2018. As expected, most news are reported for the city center area, whereas there is less density in the outskirts of the city.

Total Articles	17,075
Articles without Tags	3,539
Distinct Tags	2,837
Filtered Tags	135 (86)
Labeled Articles (Training Set 2014-2017)	11,675
Labeled Articles (Test set 2018)	2,791

Table 3.2: Summary of the extracted data from RomaToday

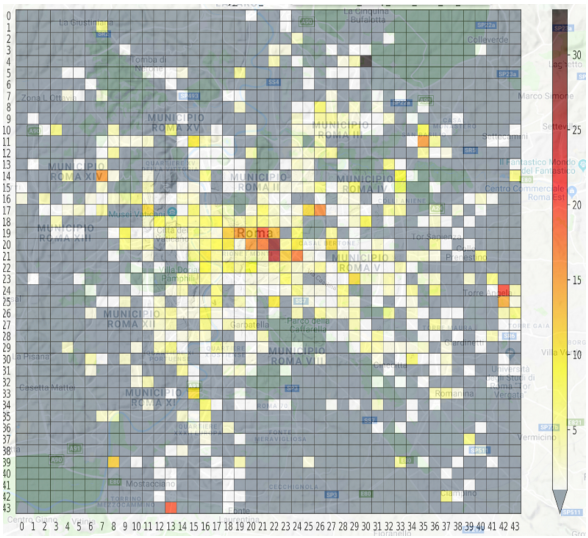


Figure 3.1: Distribution of news from RomaToday for 2018.

Identification of macro-categories

In order to identify macro categories, tags were first represented as word embeddings. Specifically, the pre-trained Italian fastText model[4] was used to this end (Grave et al., 2018). Concretely, the model was simply queried for a vector for the tag, as a string. Note that multi-word tags, such as "tentato omicidio" (attempted

[4]fasttext.cc/docs/en/pretrained-vectors.html

murder), were left as they were, without further pre-processing. Also note that, in this phase, only tags from articles in the training set were considered.

Then, agglomerative hierarchical clustering was applied to the tags embeddings. Results, in the form of dendrogram, are shown in Figure 3.2.

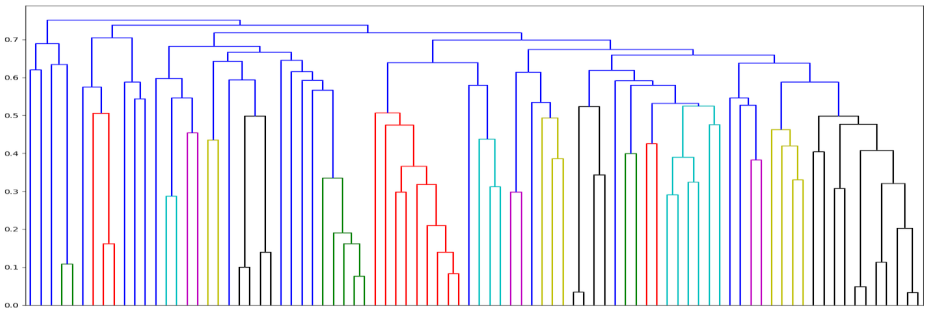


Figure 3.2: Dendrogram for clustering of article tags.

As it is clear from the dendrogram, the data distribution makes it possible to distinguish several clusters, especially when considering higher distances between the clusters. In order to obtain the actual macro-categories, several values for the dendrogram cut were experimented. We manually evaluate the obtained clusters in order to assess the value that best suited our needs in terms of distinction among categories. We finally chose a value of ≈ 0.63. By choosing this value, it was possible to observe 12 different clusters that best represent the macro-categories of news articles in the training data. In fact, by choosing a higher threshold, the clustering would have yielded too broad categories, while lower ones would have yielded a too fine grained categorization, not suitable for the purpose of this research.

Once the tags were clustered into macro-categories, a manual evaluation was performed in order to assign an appropriate label to each cluster. Table 3.3 shows each category and the respective tags.

By looking at the macro-categories, it is clear that the pre-trained embeddings, paired with clustering algorithms, can effectively distinguish words with similar meaning, that actually describe the same category. By considering the cosine as a distance metric for the clustering, the underlying properties of word embeddings and more in general of distributional space models of words, that have already been established in the literature, can be exploited to their full potential.

The more prominently featured macro-category in terms of number of different tags is the "Crimes" category. It can be argued that this may be due to two main reasons. First, among the macro-categories, it is clearly the most generic one, encompassing several different themes, such as pursuits, shootings, theft, aggression and so on. However, the data show that it does not include crimes linked to drug usage or dealing, as they form a similar but independent category. Second, it is clear that for a newspaper reporting city news, and especially in such a vast city as Rome,

MACRO-CATEGORY	TAG
Crimes	pursuit, shootings, shooting, guns, weapons, explosions, attempted murder, arrests, arrest, complaints, pickpockets, extortion, fines, House arrest avoidance, theft, muggings, investigations, car theft, murders, apartment theft, brawl, stabbings, brawls, robbery, aggression, thief apartments, frauds, quarrels, seizures, familiar quarrels, thieves, robberies, threats, copper theft, vandalism
Events	demonstration, protests
Squatting	commercial squatting, illegal sellers, illegal settlements, squatting
Violence against Women	women violence, domestic violence, sexual violence, harassment, rapes, stalking
Drugs	drug arrests, hashish, spaccio, drug dealing, drug, cocaine, drugs, pusher, prostitution, marijuana
Terrorism	terrorism, bomb alert
Environmental problems	fires, fire, bad weather, flooding
Minorities and diversity	shanty town, evictions, nomad camp
Roads and Traffic	streets, road checks, road works, accidents, investment, traffic, road closed
Urban Decay	holes, garbage, decay, collapses, fallen trees
Suicides	suicides, attempted suicide
Missing People	missing people

Table 3.3: Macro-categories and their respective tags

crime reports are expected to be featured predominantly with respect to other categories, and to be described with several different, yet very similar among each other, tags.

In order to get an idea about the overall distribution of macro-categories in the dataset, each article of the training set was labelled with a macro-category according to its tags. In order to be as accurate as possible with the labelling, in this case all the articles without tags, and articles with tags associated with more than one category, were considered as unlabelled. In total, we obtained 14,466 labelled articles. The distribution of categories obtained in this way is shown in Figure 3.3. As expected, the vast majority of articles fall under the *Crimes* macro-category, also due to the high number of tags. Other highly represented categories in the dataset are *Roads and Traffic*, *Environmental Problems*, and *Drugs*. On the other hand, the least represented categories are *Missing People*, *Events*, and *Terrorism*.

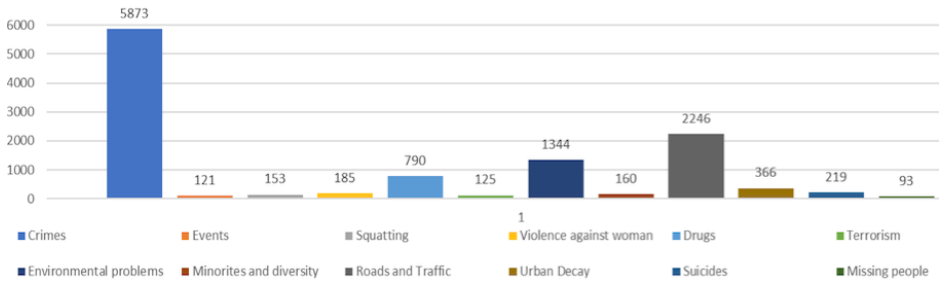


Figure 3.3: Distribution of macro-categories in the training set.

Text categorization model

After evaluating the quality of the macro-categories in terms of tags they contain, and having generated the labels for the training set, the text categorization model is trained on the available articles. The problem is in this case formulated as a multi-class classification problem. Specifically, given an article, represented by its text (i.e. title and body), the goal is to label it with one of the 12 predetermined macro-categories obtained with the tag clustering. Therefore, the problem is considered a 12-class classification task.

In order to find the best trade-off between accuracy and computational cost, several experiments concerning both the feature space and the learning algorithm have been performed. For the feature space, as previously mentioned, two options were considered to build the bag-of-words representation of news. On the one hand, a *complete* representation taking into account the entire text of each piece of news, and on the other hand a *partial* representation that only considers the title and the summary. The complete representation clearly retains more information in its bag-of-words vectors, but at the cost of sparsity and computational cost, while using the partial representation may yield a more dense representation and be computationally economical to exploit. As for the learning algorithm, experiments were performed with *Logistic Regression* (LR), *Decision Tree* (DT), and *Support Vector Machine* (SVM). Thus, overall 6 different models were built. Each model was first evaluated via 5-fold cross validation to assess its performances. Note that the training data contain news articles from 2014 to 2017, while the 2018 data were used for testing. Again, the class (i.e. the macro-category) assigned to both training and test sets is given by the cluster containing the tags for each article. Articles without tags or associated with more than one macro-category were discarded. In total, the training set consist of 11,675 news articles. Results for each model on 5-fold cross validation are reported in Table 3.4.

For both the complete and partial representations, results show that the classifier based on SVM outperforms the other classifiers by a wide margin. Overall, the best results have been obtained with the *SVM_Complete* model. However, the model

Classifier	Accuracy	Precision	Recall	f1-score
LR_Partial	0.84	.092	0.42	0.49
DT_Partial	0.84	0.71	0.69	0.70
SVM_Partial	**0.90**	**0.87**	**0.73**	**0.79**
LR_Complete	0.87	0.93	0.51	0.59
DT_Complete	0.85	0.71	0.69	0.70
SVM_Complete	**0.93**	**0.90**	**0.79**	**0.84**

Table 3.4: 5-fold cross validation on different models and feature spaces.

ultimately implemented in the framework was the *SVM_Partial* one. This choice was motivated by the fact that, albeit performances for the *SVM_Partial* model are slightly worse, especially in terms of recall, than those of the *SVM_Complete* one, it has a clear advantage in terms of both training and prediction times. This is because the feature space, i.e. the vocabulary used for the bag-of-words representation, is three times smaller. More specifically, the feature spaces for the *SVM_Complete* and *SVM_Partial* models are respectively 75,000 and 24,000 dimensions. This reflects on the training time as well. For training the *SVM_Complete* model on a laptop with a 2.6 GHz Quad-Core Intel Core i7 and 16GB of RAM, it took around 70 seconds. The *SVM_Partial* model was trained in around 10 seconds instead. Class-wise performances for the *SVM_Partial* model are reported in the confusion matrix in Figure 3.4. As expected, the model perform worse for under-represented classes, such as "Squatting", "Violence against women", and "Events". However, we can argue that most such under represented classes actually fall under the broader umbrella of the "Crimes" macro-category (e.g. "Terrorism", "Violence against women"). It is also interesting to note how also "Events" articles are often labelled as "Crimes". This may be due to two reasons. First, big events in the city are expected to be a catalyst for criminal activities. Second, if we look at the tags considered for the "Events" category, we can note that protests and demonstrations are considered as events. It is possible that articles describing such events are in fact similar to articles describing crimes, by for example citing a police intervention.

Finally, the evaluation on the test set was performed. The test set includes a total of 2,791 articles from 2018. In order to obtain gold labels for the test set, the same procedure applied for training was followed. Again, the label is assigned based on tags. Articles with no tags or with tags pointing to different macro-categories were discarded.

Two different analyses were performed on the test set. First, the whole training set was used for learning. In this case, the system obtained an F1-score of 0.79 (Precision 0.84, Recall 0.75), and an Accuracy of 0.91. Second, as the framework is expected to perform in real-time streaming conditions, an additional experiment

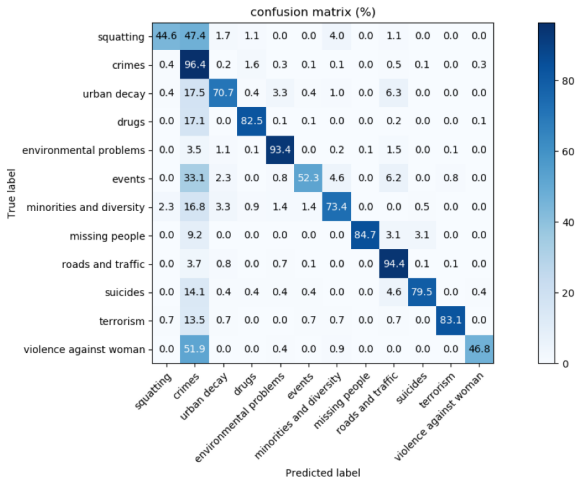


Figure 3.4: Confusion matrix for SVM_partial, 5-fold cross validation.

was performed. Specifically, it was chosen to test the algorithm by incrementally adding more data to the training set. The model was first trained on data from 2017, and tested on 2018 articles. Then, data for each previous year were incrementally included in the training set in order to evaluate the performances as new data are provided to the algorithm. Results for the whole dataset and for each run with incremental data are reported in Table 3.5.

Training data	**Accuracy**	**Precision**	**Recall**	**f1-score**
2017	0.88	0.80	0.62	0.68
2017+16	0.90	0.80	0.68	0.72
2017+16+15	0.91	0.84	0.70	0.75
2017+16+15+14	**0.91**	**0.84**	**0.75**	**0.79**

Table 3.5: SVM_partial performances on test set with incremental training sets.

As expected, the model trained on the whole training set performed best. Performances in terms of F1-score grow almost linearly with the size of the dataset, as shown in Figure 3.5. Particularly interesting is the improvement on the Recall. It can be argued that two factors play a key role in this regard. First, as the algorithm is fed more data, its generalization capabilities are expected to improve. Second, the addition of new data implies an increase in the feature space used for representing text, as the vocabulary is expected to increase. This behaviour is similar to the one reported for the complete and partial models. In fact, the complete model had a wider margin of improvement especially on the recall, with respect to the partial one. Albeit the F1-score improves considerably by increasing the size of the dataset, it must be noted that the same cannot be said of the accuracy. In fact, improvement

in accuracy when training on only the prior year or a 4-year span is rather limited. This is probably due to the fact that the overall distribution of macro-categories of news articles is very similar throughout the years, with no noticeable differences or new macro-categories to be reported. Nonetheless, accuracy results are rather encouraging.

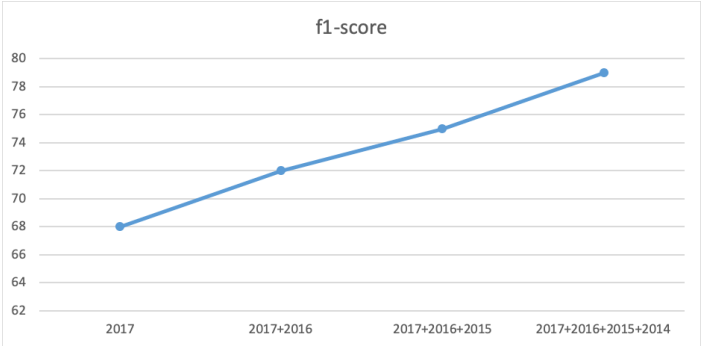


Figure 3.5: SVM_partial F1-scores on the test set using different training sets.

As a proof of concept and example of the obtained results, a use case of the framework, integrated with the Today news source and the text categorization model, is presented in Figure 3.6. Within the framework, the user can select a specific time span and zoom on a specific zone of the city and check, for each cell, the distribution of macro-categories. In the figure, the first five most frequent macro-categories of news concerning the areas of the Termini and Tiburtina Railway stations. By looking at the distribution, it could be assumed for example that these areas have issues concerning general crimes and especially drug dealing.

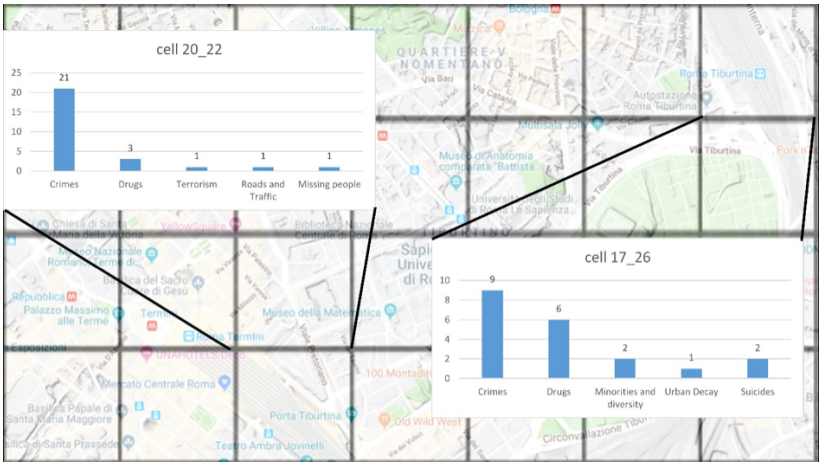


Figure 3.6: A map zoom on the city centre, including Termini and Tiburtina railway stations.

3.3 Discussion on the results

Several important observations can be made given the proposed goals and the obtained results.

First and foremost, it is clear that exploiting NLP techniques also in contexts such as smart cities research can provide an added value to the output, especially when considering as source unstructured or semi-structured data, such as news articles on the web and their metadata. City profiling is an important issue nowadays, as cities become more and more the center of activities for most people. Thus, enabling such kind of analysis may provide great benefits to frameworks such as the one taken into account for the present analysis.

Second, an evaluation on the use of clustering techniques for word embeddings was provided. While the semantic similarity or relatedness may be better modelled with different, more controlled frameworks such as for example WordNet (Miller, 1995; Fellbaum, 1998), pre-trained word embeddings have been proven to provide a reliable and efficient estimation of words meanings and how they relate to each other. The main advantage in exploiting pre-trained models is that, aside from loading the model in memory, no or little further computation is needed for generating the embeddings. Moreover, as the proposed task in rather generic, and do not hinge on context-sensitive or domain-specific information, a pre-trained model may prove to be the best choice in terms of both computational cost and quality of the results. However, it must be noted that word embedding models, and especially pre-trained ones, albeit very effective in representing the meaning of single words, or at most elements such as noun phrases, may not turn to be as reliable for entire sentences or documents. This specific issue is discussed and analyzed in detail in Chapter 4.

Third, experiments have shown that SVM are a viable strategy for classification of news in macro-categories identified in a semi-automatic fashion. Albeit in recent years most natural language understanding tasks have been performed with neural network models, it can be argued that a more traditional approach, both in terms of learning algorithm and representation of the features of text can achieve rather encouraging results. There is a vast amount of literature regarding SVM models as some of the best performing models in NLP tasks, and the obtained results definitely corroborate this fact. Nonetheless, as a potential future direction, it would be definitely interesting to experiment also with more modern architectures such as Transformers (Devlin et al., 2019), that have obtained state-of-the-art results in most NLP classification tasks via the pre-training and fine-tuning paradigm. This is also true of word representation. Albeit research on this topic has demonstrated that word embeddings based on Skip-Gram or D-BOW are able to outperform Transformer-based ones in certain tasks, it would be nonetheless interesting to evaluate their performances on this rather simple, yet still challenging, task.

Finally, given the obtained results, it is clear that the natural direction of this kind of research, especially in terms of distributional representation, is provided by the possibility of modelling not only words, but entire sentences and documents in a distributed space. In the present case study, for example, it could allow avoiding metadata such as tags in order to directly focus on the data itself, and identify similarities and regularities among articles that can lead to their profiling in a strictly unsupervised way. This is advantageous especially because it eliminates the requirement of labelling data and learning complex supervised models that could, in time, lose their generalization capabilities due to shifts in the concepts and notion that are provided in the data. This issue is at the core of the discussion presented in Chapter 4.

3.4 Summary

In this Chapter, an approach to the profiling of news articles in macro-categories in the context of a framework for profiling city areas was presented.

Section 3.1 describes the proposed methodology. The framework for city areas profiling is first presented. Then, a description of how online newspaper can be used as data source is proposed. Subsequently, a description of the two key elements is shown. On the one hand, a method for obtaining macro-categories based on article tags word embeddings. On the other hand, a text categorization model aimed at classifying news articles in one such macro-category.

In Section 3.2, the experiments performed to evaluated the proposed methods are presented, including the collection of the dataset. Experiments consist in (i) identifying macro-categories from tags of real world articles via clustering, and (ii) training the text categorization model to label articles with macro-categories, with training data of varying size.

Finally, Section 3.3 proposes a discussion of the obtained results and potential further works in this area.

Chapter 4

Using distributed representation to identify professional figures from résumés

Modelling sentence and document similarity via unsupervised learning algorithms enables the possibility of identifying patterns that can be exploited for profiling without the need for additional structured information that is not directly included in the text. In this chapter, such idea is explored by considering a case study concerning the profiling of professional figures through the analysis of job applicants' résumés.

Profiling professional figures is becoming more and more crucial, as companies and recruiters face the challenges of the so called *Industry 4.0*. Nowadays, the recruiting process of any company holds an important strategic and economic value. In fact, the identification of the best candidate for a given job, and the identification of candidates with diverse skills that can work well with each other is a valuable asset for any company. It can be argued that this holds especially true for sectors such as Information Technology (IT), where the market is extremely dynamic, and specific professional profiles often have a short time span, linked to the most trending technologies. These are in fact often proposed and replaced within a few years, forcing professionals to dynamically update their knowledge in order to be suitable for different job positions, and remain appealing in the job market. These considerations are valid and current for any work sector that has been influenced by Industry 4.0.

Both recruitment agencies and Human Resource (HR) departments within companies are therefore in need of modern tools that can help them in the process of recruiting new professional figures or re-assigning resources to different departments based on their skill set. Such tools are often referred to as *Applicant Tracking Systems* (*ATSs*). These systems have been extensively used in HR departments and recruit-

ment companies to track resources and skills. Most existing ATSs are based on the *Customer Relationship Management* (*CRM*) paradigm, but with a specific focus on the tracking of potential candidates and their characteristics (e.g. experiences, availability, and skills), and the identification of the most suitable candidates for specific job opportunities. Such systems are widely adopted and provide good performances, both in terms of matching and computational costs.

Today's ATSs belong to two main categories: *direct recruitment* and *indirect recruitment*. Direct recruitment ATSs are specifically targeted for HR departments, in order to perform the direct recruitment of resources for the various departments and jobs available internally in the company. They are often also used to keep track of employees and their skills. Indirect recruitment ATSs are instead geared towards consulting, recruiting and interim agencies. The specific needs of such companies are in fact different, and cover a wider range of tasks. For example, they must be able to store and match job opportunities and potential candidates coming from different sources. Moreover, they need to be reliable, efficient, mostly autonomous, and more generally user-friendly, in order to simplify the work of recruiters and similar professional figures.

One of the main drawbacks of ATSs in general is that they are often based on manual evaluation of résumés and heuristic rule-based algorithms in order to find the most suitable candidates. Therefore, in the last few years, there has been an increasing interest, both from an industrial and a research perspective, in developing data-driven tools that can automatically assist recruiters and personnel managers in their daily life. Ideally, such systems should be able to automatically identify similar profiles, based on information that can be extracted from their résumés.

Albeit the topic is currently attracting an increasing interest both from a research and an application standpoint, literature concerning especially text mining in this specific domain is still scarce. Very few researches on the impact of ATSs on the recruitment process have been proposed within the business economics field (Eckhardt et al., 2014; Laumer et al., 2014). In addition, a small number of recent approaches have proposed the implementation of data mining and information retrieval techniques both in the recruitment process and in performing job recommendations (Heggo and Abdelbaki, 2018; Shehu and Besimi, 2018). We can argue that this may be due to two key factors. First, résumés often contain private information about candidates, such as phone numbers, addresses and so on. Due to privacy concerns, especially in the research field, it is becoming increasingly difficult to collect and store such kind of data and comply with GDPR-like regulations. Second, it is even more difficult to obtain such data with gold labels (i.e. a ground truth about the résumé) for specific tasks, such as document classification or categorization. It is clear that extensive research in the direction of improving software such as ATS is still in its embryonic stage.

In the present work, two different approaches focused on building and analyzing distributed representations of résumés are proposed. The approaches are rather similar in their premises, in that they both exploit unsupervised algorithms to learn representations of sentences and entire texts.

The first approach employs an algorithm for keyword extraction based on *word entropy* weighting over time (Dumais, 1992). Such keywords are used for learning distributed representations of the résumés via the *paragraph vector algorithm* (Le and Mikolov, 2014). The representations are evaluated through manual analysis and clustering techniques. The approach is presented in Sec. 4.1. Such approach, albeit effective, presents several important drawbacks. First of all, it results to be quite computationally expensive, as résumés are passed through several demanding steps of analysis, including pre-processing, keyword extraction, and the learning phase by means of Doc2Vec. In a real world scenario, all such steps should be repeated as new data are collected. Second, concerning the keyword extraction phase, albeit effective, it may produce spurious results that, in turn, could affect the learning phase and thus the quality of the final representation. Finally, albeit Doc2Vec is shown to effectively learn from domain-specific data, and perform better than simple pre-trained word embeddings models, it is clear that in order to learn accurate representation a lot of such domain-specific data are required, that may not be available.

The insights gained from the analysis of the performances on the first method and its key drawbacks drove the development of the second approach, presented in Section 4.2. In order to remove the need of searching for specific keywords, summarization techniques are proposed that effectively remove redundant information often contained in résumés. Then, to address the issues generated by using Doc2Vec as a language model, the use of Sentence-BERT (Reimers and Gurevych, 2019) is proposed. Sentence-BERT is a state-of-the-art pre-trained language model based on the transformer architecture that is specifically tuned for sentence-level representation. Also in this case, the quality of the resulting representation is evaluated through clustering techniques, both quantitatively and qualitatively.

As for the clustering, *agglomerative hierarchical clustering* was chosen to evaluate and profile résumé embeddings in both approaches. One of the main reasons to choose hierarchical clustering over other different partitioning algorithms is that hierarchical clustering yields a clustering that has an intrinsic hierarchy, based on a distance metric between objects, that can be explored in both directions. This may be crucial when considering data such as résumés, which have to be grouped into professional profiles. In this way, such profiles can be described at different levels of granularity, from broader to more specific ones. This may prove invaluable when dealing with new, unseen résumés, that can therefore be assigned to a profile based on the hierarchy, at various levels of granularity. Agglomerative hierarchical clus-

tering algorithms seek to build a hierarchy of objects based on a linkage criterion that is used to merge two clusters of objects (also containing one object only) into a unique cluster at a higher level of the hierarchy. The algorithm follows a bottom-up approach. At the beginning, each object belongs to one cluster. Iteratively, at each level of the hierarchy the algorithm merges the two most similar clusters based on the linkage criterion, until all objects are assigned to one cluster.

From the obtained results, it is clear that the transformer-based architecture, that employs pre-trained models, is able to better model longer sequences into semantically relevant representations, proving that transformers with strong pre-training are rather effective in tasks and scenarios where the meaning of sentences and entire texts must be compared and modelled.

4.1 Résumés as Bag-of-Keywords with Doc2Vec

The first described approach exploits more traditional techniques of NLP and distributional semantics to identify profiles from résumé texts. An overview of the entire process, including a clustering phase in which distributed representations of résumés are analyzed for profile identification, is shown in Figure 4.1.

In the NLP phase, two different steps are applied to the data. First, a keyword extraction algorithm is used to identify and extract potential candidate keywords, both single and multi-words, from each résumé, by employing *term entropy* as a weighting measure for their relevance. Further, such keywords are exploited to generate a distributional semantic model of résumés by means of the *paragraph vector* algorithm (Le and Mikolov, 2014).

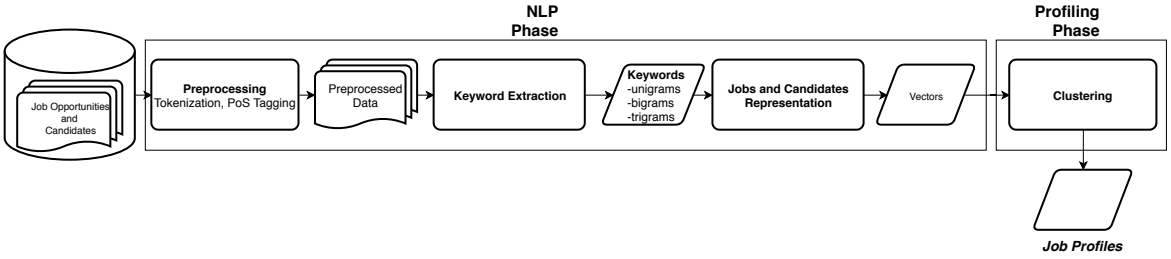


Figure 4.1: Overview of the approach based on keyword extraction and the paragraph vector algorithm.

The data are first pre-processed in order to both improve the quality of the obtained representation and reduce the computational cost. The goal is to obtain content words from texts and filter out several words that are not interesting for the analysis. In order to do so, *sentence splitting*, *tokenization* and *Part-of-Speech* (*PoS*) *Tagging* are applied to each résumé using SpaCy,[1] a Python toolkit which provides

[1] https://spacy.io/

deep learning based models to perform linguistic analysis for several languages, including Italian, out of the box. In addition, we remove a list of *stop-words* such as "Curriculum vitae", that are not interesting for the representations.

Keyword Extraction

For the keyword extraction algorithm, two main goals are pursued. On the one hand, keyword quality is taken into account. The aim is to identify keywords that can best describe résumés without losing information, and thus improve the profiling. On the other hand, it is interesting to take into account time, i.e. when a given résumé was actually produced or used to apply for a position. Such temporal aspect may prove to be beneficial for the identification of regularities and novelty in word usage. Such novelty may in turn represent the emergence of a novel profile or professional figure.

Figure 4.2 shows the flowchart of the keyword extraction algorithm. The algorithm itself is rather simple and straightforward. A time-window strategy based on the production date of résumés is adopted for selecting batches of data, in order to identify the most relevant keywords for a given period of time. The time window can be set and is in the range $[1, n]$ days.

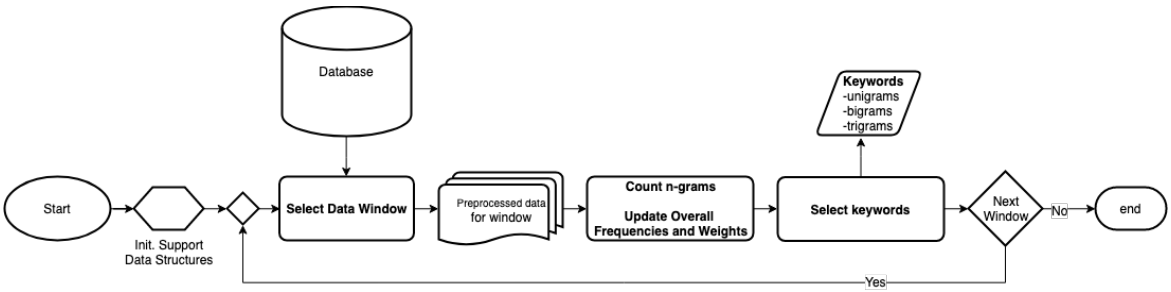


Figure 4.2: Flowchart of the Keyword Extraction algorithm.

The main idea is to identify words and terms (i.e. *n-grams*) by exploiting their frequency in order to compute a set of weights. The weights are used sequentially to identify candidate terms and filter them out. First, for each time window the frequencies of the n-grams with respect to each document are computed. N-grams are considered a sequence of n sequential tokens in the text. While the algorithm allow for varying sizes of n, in the proposed implementation only n-grams up to tri-grams are considered. Moreover, only certain PoS patterns are considered to be relevant. For instance, we want to identify n-grams such as "full stack developer", but we have no interest in n-grams like "expert in", that would increase the computational complexity of the algorithm without extracting meaningful terms. Specific content word PoS (e.g. *nouns*, *adjectives*) and specific PoS patterns that are often found in

multiword expressions are considered, such as *adjective-adposition-noun*, as in "fluent in Python".

Then, the relative frequencies computed for the specific time window are used to perform an on-line update of the overall frequency counts and weights for each n-gram.

Concerning the weights and selection criteria for the identification of keyword n-grams, several cascade selections and weightings are performed. In order to identify candidate n-grams, we exploit classical association measures such as *Pointwise Mutual Information* (PMI) (Church and Hanks, 1989). The PMI of two or more words quantifies the discrepancy between their joint probability and their individual marginal probability by assuming independence. More specifically, given two words x and y,

$$PMI(x;y) = \log \frac{p(x,y)}{p(x) * p(y)}$$

Probabilities are estimated by means of frequency. In particular, $p(x)$ (and $p(y)$) can be easily estimated as $f(x)/N$, where $f(x)$ is the frequency of x in the data, and N the total number of unigrams. Joint probability $p(x,y)$ is computed as $p(x,y) = f(x,y)/N_{bigrams}$.

In addition, since PMI is based on joint probabilities of events, it satisfies the *chain rule* (or general product rule) of probability. Thus, PMI for trigrams is computed as $PMI(x;yz) = PMI(x;y) + PMI(x;z|y)$. We implement a slight variation of the PMI formula, namely *Positive PMI* (PPMI). PPMI is simply defined as $PPMI(x;y) = [PMI(x;y)]_+$. We select as candidate n-grams only terms that have a PPMI equal to or larger than 1.00. This ensures that we only select pairs or triplets of words that have a chance of co-occurring together higher than they were independent. Note that PPMI for unigrams is 1.00.

Once candidates terms are identified, a scheme based on term entropy was selected to finally extract the keywords. Term entropy (Shannon, 1948) measures the average uncertainty of a given term in a collection of documents (Dumais, 1992), thus providing an efficient method for estimating the relevance of such term for the collection of documents at hand. It is defined as:

$$e(t) = 1 + \sum_{j=1}^{D} \frac{p(t_j)\log(p(t_j))}{log(D)}$$

where D represents the overall number of documents in the collection, and $p(t_j)$ represents the probability of the term t for the document j. Again, such probability can be approximated by looking at term frequencies. More specifically, if $f(t_j)$ is the frequency of term t in document j, then $p(t_j) = \frac{f(t_j)}{\sum_{j=1}^{D} f(t_j)}$.

In summary, the proposed approach updates PPMI and entropy for all considered terms, and selects those with a PPMI equal or larger than 1 and an entropy value equal to or larger than a predetermined threshold WT for each iteration (i.e., for each time window).

By employing such strategy, both terms that frequently occur across all time windows, and novel terms that were previously not considered, can be identified. Moreover, a set of keywords that have been manually selected as relevant by experienced recruiters are employed as well. In this framework, both manually selected keywords and automatically extracted ones coexist.

Each résumé is actually represented by the keywords that have been considered relevant for its time window that are found in its texts. The advantage of this representation is twofold. First, considering only relevant keywords allows learning a higher quality representation of texts. Second, as relevance of keywords is recomputed for each time window, this could ensure a greater ability to discriminate between different skills represented in résumés and job opportunities.

Résumé representation with Doc2Vec

Once each résumé is represented with its relevant keywords, *Doc2Vec* or the *paragraph vector algorithm* (Le and Mikolov, 2014), is used to generate résumé level embeddings. The algorithm builds upon the notion of word embeddings by first learning word embedding representations, and then using them in order to generate a vector for the whole document.

Two different architectures are available for modelling sentences and documents with Doc2Vec, namely *Distributed Memory* (PV-DM) and *Distributed Bag-of-Words* (PV-DBOW). The former exploit a concatenation of paragraph vectors and word vectors with the objective of predicting the next word in a given window. The latter simply learns vector representations by means of predicting words sampled at random from the output paragraph. As résumés have a semi-structured format, in which similar patterns of words are often repeated among résumés, a language model that takes into account word ordering such as PV-DM may be easily fooled in assigning a higher similarity score to different profiles due to very similar lexical and syntactic structures. Thus, PV-DBOW was chosen as a learning algorithm for résumés.

Evaluation of the approach

Several experiments have been performed to assess the effectiveness of the approach. Experiments were executed on a dataset of 12,957 Italian résumés. The résumés were made available by the company IT Partner Italia following the General Data Protection Regulation (GDPR). They describe profiles in the IT sector, and were

produced between 2016 and 2017. Each résumé is provided with several additional annotations, such as city of provenance and skills of the candidates (i.e. main skill, secondary skill, etc.). Such information is however automatically extracted from the résumés using a proprietary algorithm, and therefore cannot be considered as a gold labelling. However, it can still provide a rather good approximation of the skills of a given candidate, that can be in turn used for the evaluation. It must be observed that, as résumés were obtained from different sources and in different formats, errors and inconsistencies could arise during conversion in plain text.

The following parameters were adopted for the keyword extraction algorithm. For the PoS patterns, we included nouns, proper nouns and adjectives, and all combinations of three nouns, proper nouns, adjectives, adpositions and determiners, that form multi-word patterns in Italian, e.g. "dispositivi Android" (Android devices), "reparto tecnico" (technical department), "sviluppatore Java" (Java developer). For the time window, 180 days was chosen, as a too small window would yield results that are too fine grained for the proposed goal, and a broader one could hinder the performances regarding time-specific keywords.

For what concerns the PV-DBOW algorithm, the implementation provided in Gensim was used, named Doc2Vec.[2] The language model was trained for 5 epochs, with a minimum number of occurrences per word equal to 5. The output vector size was set to 200. All the other parameters and hyperparameters were left to their default value.

Finally, for the sake of comparison, a representation based on pre-trained word embeddings was also implemented. The pre-trained model of *fastText* for the Italian language was used (Grave et al., 2018). In this case, the résumé embedding was simply computed as the mean of word embeddings for the keywords.

The available dataset lacks appropriate gold labels. Thus, the proposed evaluation is performed in order to determine the capability of the system in distinguishing between rather similar and very different profiles. For example, we aim to represent a Java developer and a database engineer with vectors that are farther away with respect to those representing a Java developer and a Python developer résumés. In order to do so, two small sets of résumés were selected based on their main skill, and evaluate the text of the résumé to verify that information. Such skill could represent at least a reasonable approximation of the profile of the résumé. As the dataset is not publicly available, no other information can be provided in this context. The first set includes résumés that are labelled with the same main skill "Java Junior". The second set includes instead résumés that describe very different skills, that are "Java Senior", "Help Desk intermediate", "Network engineer junior", "Web designer senior", and "Accounting employee".

Both the Doc2Vec and fastText representations are evaluated by means of co-

[2]https://radimrehurek.com/gensim/models/doc2vec.html

sine similarities between the learned representations. Results for the two sets and for the two representations (Doc2Vec and pre-trained word embeddings) are reported in Tables 4.1 and 4.2. The proposed approach based on Doc2Vec appears to be promising, as résumés for the various "Java Junior" profiles tend to have a higher cosine similarity among them, while similarities between different professional figures are generally much lower. On the contrary, the pre-trained fastText representation yields cosine similarities that are very high between all the résumés. This may occur because by simply averaging pre-trained word embeddings of rather complex texts, it is likely that the resulting vectors will be actually found in the same region of the space. This may also be due to the fact that, in the case of traditional pre-trained word embeddings such as fastText, they are actually trained on general-purpose corpora (e.g. Wikipedia). On the contrary, the Doc2Vec representation is learned directly from domain-specific data. It is clear that a more refined analysis would be in order to better evaluate the method. Nonetheless, it is useful to notice such difference at a glance.

ID	1	2	3	4	5
1	1	#	#	#	#
2	0.65	1	#	#	#
3	0.66	0.96	1	#	#
4	0.70	0.95	0.94	1	#
5	0.51	0.53	0.58	0.61	1

Main Skill	Java Sr.	Help Desk	Network Eng jr.	Web Design sr.	Acc.
Java Sr.	1	#	#	#	#
Help Desk	-0.01	1	#	#	#
Network Eng jr.	0.35	-0.01	1	#	#
Web Design sr.	0.43	-0.10	0.50	1	#
Account. Employee	0.60	-0.08	0.27	0.52	1

Table 4.1: Cosine similarities between *Java Junior* profiles (left) and between different profiles (right) using Doc2Vec.

ID	1	2	3	4	5
1	1	#	#	#	#
2	0.77	1	#	#	#
3	0.91	0.78	1	#	#
4	0.91	0.83	0.97	1	#
5	0.91	0.84	0.96	0.97	1

Main Skill	Java Sr.	Help Desk	Network Eng jr.	Web Design sr.	Acc.
Java Sr.	1	#	#	#	#
Help Desk	0.96	1	#	#	#
Network Eng jr.	0.96	0.96	1	#	#
Web Design sr.	0.93	0.91	0.91	1	#
Account. Employee	0.93	0.97	0.94	0.91	1

Table 4.2: Cosine similarities between *Java Junior* profiles (left) and between different profiles (right) using fastText pretrained word embeddings.

Finally, in order to verify whether the approach can enable the identification of profiles and discriminate among résumés with diverging skill sets, hierarchical

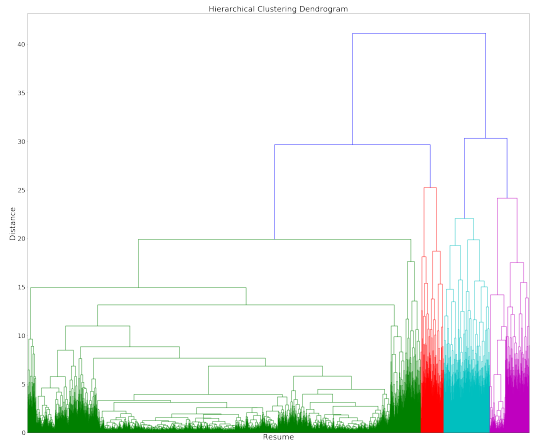


Figure 4.3: Doc2Vec representation clustering.

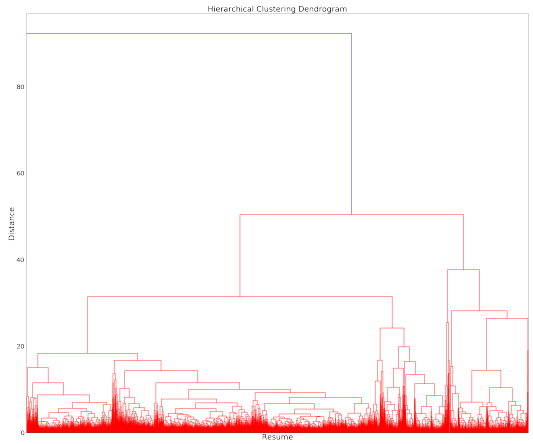


Figure 4.4: Pre-trained embeddings clustering.

clustering was applied to both the Doc2Vec and fastText-based representations of résumés. More specifically, a complete-linkage hierarchical clustering with cosine as distance metric was used. Fig. 4.3 and Fig. 4.4 show the two dendrograms obtained by the two representations, respectively. By analysing the dendrograms, it can be noted that different clusters are actually determined, albeit not clearly distinguished. The Doc2Vec representation appears to yield more well-defined and homogeneous clusters with respect to the pre-trained fastText. We can appreciate how actually different clusters of curricula can be identified. It is evident that the approach proposed is able to determine groups of curricula with similar characteristics. For instance, if we cut the dendrograms at distance 10, we can observe that for the pre-trained embeddings representation several clusters contain less than 10 elements, and two macro clusters contain a number of résumés up to around 3500. Conversely, the clustering produced with Doc2Vec at the same distance is more homogeneous. Résumés and keywords are more evenly distributed across clusters.

Further, a number of clusters for the two representations was analyzed. We realized that the clusters generated by using pre-trained word embeddings are generally less relevant for a recruiter. Just to give an example, using Doc2Vec we obtained for example a cluster with frequent keywords such as “web service”, “TCP IP”, “Active Directory”, “MYSQL”, “web application”, that clearly describe the profile of a *web developer*. On the other hand, by analysing clusters obtained with pretrained word embeddings, they appear to be less descriptive. For example, one cluster contains the following frequent keywords: : “C”, “Java”, “Photoshop”, “Google Analytics”, “e-mail”, “marketing”, “assistente amministrativo” (administrative assistant). It is clear that such cluster cannot be considered as a proper representation for a given profile.

The approach proposed in this section is promising, but it presents several im-

portant drawbacks that should be taken into account. First, it is clear that the approach may be quite computationally expensive, as résumés are passed through several steps of analysis, including pre-processing, keyword extraction, and the learning phase by means of Doc2Vec. In a real world scenario, all such steps should be repeated as new data are collected. For example, for a dataset consisting of 10,000 résumés, the processing time approaches 8 hours of computation on a virtual machine equipped with a 2.6 GHz Dual-Core Intel Core i5 and 6 GB of RAM. Second, concerning the keyword extraction phase, albeit effective, it may produce spurious results that, in turn, could affect the learning phase and thus the quality of the final representation. Finally, albeit Doc2Vec can effectively learn from domain-specific data, it is clear that in order to learn accurate representation a lot of such domain-specific data are required, that may not be available. On the other hand, traditional pre-trained word embeddings models appear to be not suitable for modelling the semantics of sentences and documents. Therefore, a different strategy is proposed in order to address such issues and improve performances.

4.2 Summarization and pre-trained Language Models for categorization

Given the shortcomings pointed out for the approach presented in 4.1, a second method has been proposed to enable the detection of profiles of professional figures from résumés.

On the one hand, the proposed approach relies on *extractive summarization* rather than keyword extraction to reduce and simplify the texts of résumés. This has the advantage that whole sentences are considered as relevant, thus maintaining both the lexical structure and the words. Moreover, it enables for reducing redundancy of information typically contained in résumés texts.

On the other hand, the use of pre-trained Transformer-based language models is proposed to represent sentences and, in turn, whole résumés, rather than Doc2Vec. This can be beneficial for several reasons. First, no further learning is needed, thus reducing the computational cost of projecting words and sentences in a distributed semantic space. However, we must note that, given domain-specific data, such models could be further pre-trained on such domain-specific knowledge to improve performances. Second, these systems, and more generally Neural Language Models (Bengio et al., 2003), allow for a representation of words and sentences that can incorporate their semantic quality, therefore enabling the identification of semantically similar words, sentences, and entire texts. In particular, Sentence-BERT (Reimers and Gurevych, 2019) models provided state-of-the-art performances in numerous sentence-level tasks that require to embed the meaning of whole sen-

tences in an n-dimensional vector and compute their reciprocal similarity.

Also in this case, agglomerative hierarchical clustering on résumé-level embeddings is applied to identify profiles, and the performances of the clustering based on the different pipelines of representation are evaluated both qualitatively, by visually analysing the outputs of the clustering algorithm, and quantitatively, by considering performance indexes such as Adjusted Rand Index (ARI) and BCubed Precision, Recall, and F1 Score. The results obtained on the available dataset are encouraging, and allow stating that the proposed approach could be viable in building systems that can identify profiles of professional figures based solely on data and in an unsupervised fashion.

Figure 4.5 present a visual representation of the pipeline for the proposed approach to obtain profiles from résumés.

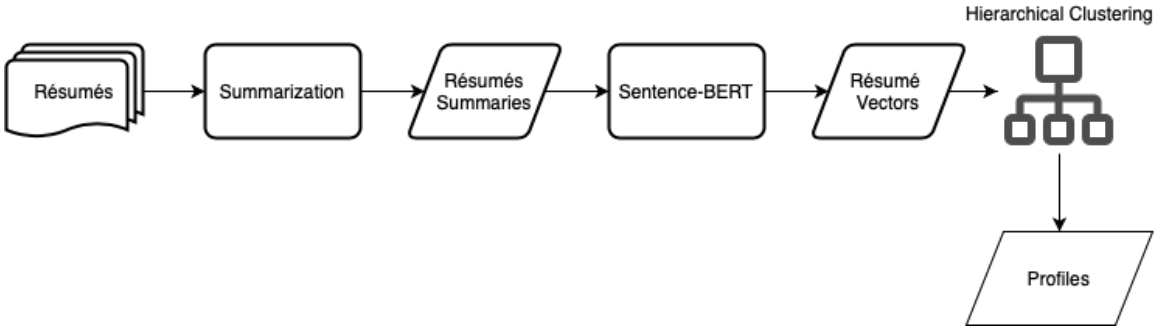


Figure 4.5: Flowchart of the approach based on summarization and Sentence-BERT.

Summarization

In order to perform summarization of résumés we experimented two different techniques, both based on the concept of sentence representation.

The first techniques relies on BM25-TextRank algorithm proposed in Barrios et al. (2016). The algorithm is a simple yet effective variation of TextRank (Mihalcea and Tarau, 2004). It implements a different distance metric to compute similarity between sentences. More specifically, it uses the BM25 (Robertson et al., 1995) ranking function to determine the most relevant and informative sentences. The BM25-TextRank algorithm is available as part of the Gensim Python library.[3] (Řehůřek and Sojka, 2010) The number of sentences to keep with respect to the whole document is set via a *ratio* parameter.

As for the second technique, summarization is performed based on the algorithm described in Miller (2019). The proposed approach leverages a BERT-based

[3]https://radimrehurek.com/gensim/

language model to extract sentence embeddings trough a standard BERT architecture, and k-means clustering to identify the most relevant ones. Specifically, once sentence embeddings are obtained, k-mean clustering is applied to identify clusters and their centroids. The value of k is inferred by the number of sentences in the document multiplied by a parameter, the ratio, i.e. a real number between 0 and 1. For example, with a ratio of 0.3, the number of k clusters will be given by the number of sentences multiplied by 0.3, therefore actually keeping 30% of the whole text. The final summary is given by sentences whose representations are closest in terms of cosines to the k cluster centroids. This enables to filter out sentences that are very similar to each other, by picking the most representative one of the group (i.e. of the cluster). The implementation of the approach is made available as a Python library.[4] As for the transformer architecture and model used, the pre-trained *DistilBERT* model (Sanh et al., 2019) was chosen due to computational constraints.

From the summarization process, a shorter but more focused version of résumés is obtained, containing only the most relevant sentences. Note that résumés are still represented as plain text.

Embeddings of résumés with Sentence-BERT

Once the summarized version of the résumés are obtained, a Transformer-based architecture is leveraged to convert such summaries into n-dimensional vectors. In order to do so, a Sentence-BERT pre-trained model (Reimers and Gurevych, 2019) is used.

In the literature, pre-trained models based on the Transformer architecture have obtained state-of-the-art results in many sentence classification tasks via fine-tuning (Devlin et al., 2019) and for language modelling. Aside from such tasks, the Transformer architecture is often used for feature generation for contextualized words and sequences embeddings. However, while the use of such word embeddings has proven to be a viable, if not better alternative to word embeddings such as those from word2vec (Mikolov et al., 2013a), the same cannot be said for sentence-wide representation. Sentence-wide representation can be obtained from such architectures, typically by considering the representation of special tokens in the sequence or by performing pooling operations on vectors of tokens in the sentence. However, BERT sentence representation may not be semantically relevant, as clearly stated by Devlin et al. (2019). Thus, they may not be suited for semantic similarity tasks in unsupervised scenarios, such as clustering. Therefore, the approach introduced in Reimers and Gurevych (2019) is followed. Here, authors propose to use *siamese* networks to fine tune BERT-based models on semantic similarity tasks, in order to produce sentence embeddings that are semantically relevant and can be compared

[4] https://github.com/dmmiller612/bert-extractive-summarizer

with measures such as cosine similarity. More specifically, their approach consists in training two identical BERT models. Each model has an additional pooling layer on top of the transformer architecture, in order to compute the sentence embedding based on words. Each model is fed by a sentence, and produces an output embedding. Then, the loss of the model is computed based on a measure of similarity between the two sentences, and the weights are adjusted accordingly during fine-tuning. The measure of similarity can be either a continuous value, or a class describing if the two sentences are similar in meaning. Models are pre-trained and fine tuned on Natural Language Inference (NLI) and Semantic Textual Similarity (STS) tasks and datasets. For semantic textual similarity, Sentence-BERT models are shown to clearly outperform standard BERT models. The implementations of Sentence-BERT and several pre-trained models are available through the *sentence-transformers*[5] Python library. Among the models available for generating sentence embeddings, *distilbert-base-nli-stsb-mean-tokens* was chosen. The base pre-trained model is described in Sanh et al. (2019). The model was fine-tuned on the composition of SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) datasets, and on the training set of STS benchmark dataset (Cer et al., 2017). As for the pooling operation, mean or average pooling is used. The advantage of using DistilBERT as opposed to a BERT model is that, despite a slight decrease in performance, it is much faster to compute embeddings, and at a fraction of the computational cost. Reported performances are in fact slightly worse than BERT models (Reimers and Gurevych, 2019). However, such architectures can be easily used without the need for GPU acceleration to generate embeddings of sentences from pre-trained models.

Evaluation of the approach

A different dataset with respect to the one employed in 4.1 was chosen for the evaluation.

The dataset contains 1219 résumés written in English. A repository with the dataset is available on the platform Kaggle.[6] Since some of the résumés appear to be corrupted or not available, the analysis was performed on a total of 1,202 résumés. Each data point is represented by the plain text of the résumé and a category label, that indicates the job sector of the résumé. In total, 25 categories are represented in the dataset. Fig. 4.6 shows a bar plot of the distribution of the categories in the dataset.

[5]https://github.com/UKPLab/sentence-transformers
[6]https://www.kaggle.com/maitrip/resumes

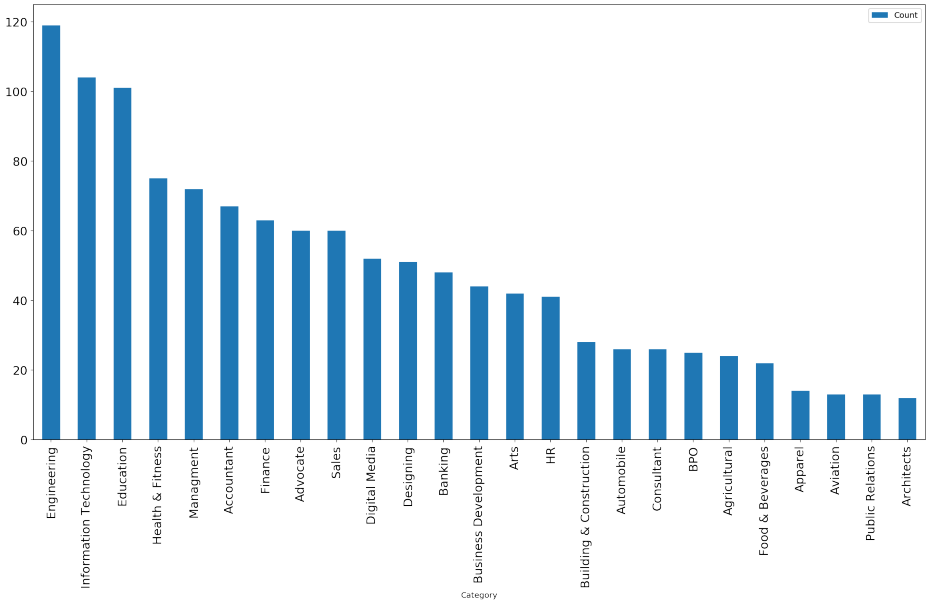


Figure 4.6: Number of instances in the dataset for each category.

Evaluation metrics

The main reason to choose such dataset, and its main advantage over the one employed in 4.1 is that, to the best of our knowledge, it is one of the only publicly available datasets with gold labels for each résumés. This enables also a quantitative evaluation of the technique. Specifically, *Adjusted Rand Index* (ARI) (Hubert and Arabie, 1985) and *BCubed Precision, Recall, and F1-score* (Bagga and Baldwin, 1998) were employed to this end.

Rand index can be used as a measure of similarity between two clusterings. If one of the two clusterings is represented by the actual labels of the dataset, it can be considered as a measure of the accuracy of the clustering algorithm under evaluation. Rand Index is computed by considering all the pairs of objects as $RI = \frac{Count\_of\_Pairs\_in\_Agreement}{Total\_Number\_of\_Pairs}$. Pairs in agreement are actually all pairs of objects that are either assigned to the same cluster, if they belong to the same category, or assigned to different clusters if they belong to different categories. As Rand Index tends to yield high values for random partitions, ARI is used. ARI is a version of Rand Index that is corrected for chance. ARI is computed as $ARI = \frac{RI - Expected\_RI}{Max\_RI - Expected\_RI}$ where expected RI is a correction for chance based on the expected similarity of all pair-wise comparisons between clusterings specified by a random model.

The BCubed metrics are based on the correctness of relationship between objects. Let $L(o)$ and $C(o)$ be the category and the cluster of object o, respectively. As stated in Amigó et al. (2009), correctness of the relationship between o and another object

o' is computed as:

$$Correctness(o, o') = \begin{cases} 1 & \text{iff } L(o) = L(o') \iff C(o) = C(o') \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Precision of o is given by the proportion of objects assigned to its cluster that belong to the same category. Recall is computed as the proportion of objects belonging to the same category as o that are assigned to the cluster of o. Overall precision and recall are computed averaging the precision and recall of each object in the dataset. Finally, F1-score is computed as $F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$.

Experiments and results

Results of the approach are evaluated both quantitatively and qualitatively via clustering analysis. As the quality of the résumé embedding is affected by the summarization method, results are evaluated by applying clustering to (i) embeddings generated after summarization with the k-means based method, (ii) embeddings generated via the BM25-TextRank algorithm, and (iii) embeddings generated with no summarization, as a baseline. For both the summarization methods, a ratio of 0.3 was chosen. Therefore, 30% of sentences for each résumés are taken into account for generating the final résumé embedding. The pre-trained model for sentence-level embeddings is the same in all experiments. Also, in all the experiments résumé embeddings are computed by simply averaging sentence-level embeddings. As for the agglomerative hierarchical clustering parameters, *complete linkage* is used as it allows creating more compact clusters than other linkage criteria, and *cosine* is used as distance metric, since it is the most widely used measure of distance or similarity among embeddings.

Figures 4.7, 4.8 and 4.9 show the dendrograms obtained with each of the proposed summarization methods.

It is already clear by a simple visual analysis of the dendrograms that the BM25-TextRank text summarization method (Figure 4.9) provides more evenly sized clusters, that better resembles the distribution of categories in the dataset.

For a quantitative evaluation of the clustering, ARI and BCubed Precision, Recall, and F1-Score were computed at different heights in the dendrogram. Specifically, cuts of the clustering were performed at distances ranging from 0.05 to 1.00 at intervals of 0.05. Results for the best four clustering obtained in this way in terms of F1-score are reported in Tables 4.3, 4.4, and 4.5.

Results show that the BM25-TextRank algorithm for summarization performs best. On the contrary, the k-means based approach with DistilBERT embeddings appears to perform worse than the baseline with no summarization. This can be

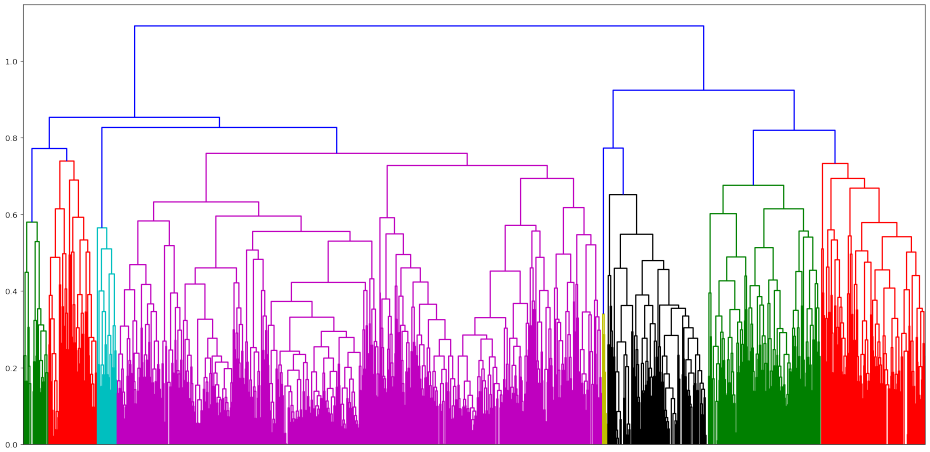


Figure 4.7: Dendrogram with no summarization (whole résumé embeddings). Baseline.

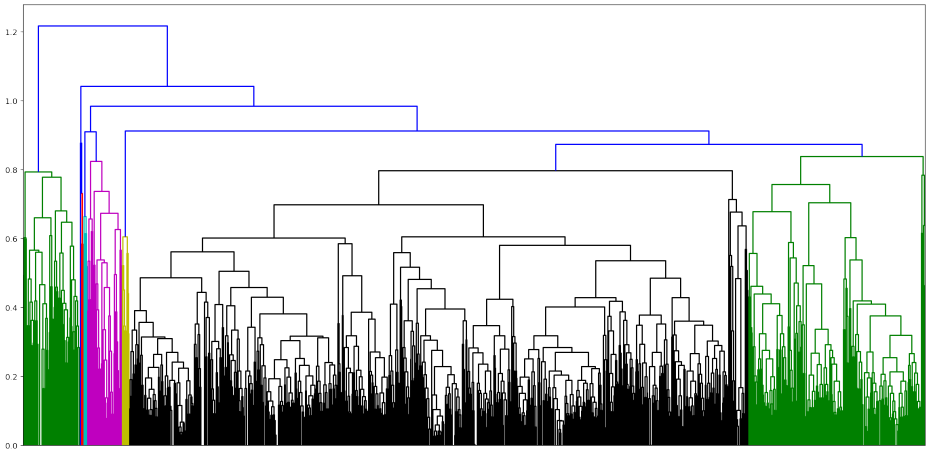


Figure 4.8: Dendrogram with K-Means summarization method.

taken as a further indication that pre-trained BERT based embeddings are not effective in generating embeddings for whole sentences or documents.

It is nonetheless important to note that obtained results are not excellent. This could be attributed to how an agglomerative hierarchical clustering algorithm performs when the density of clustered objects is different. In that case, a cut of the clustering at a given height can identify both low density cohesive clusters and at the same time clusters generated by merging high density cohesive clusters. Thus, in the case of non uniformly distributed data in the feature space, as it is the case for high-dimensional embeddings spaces, cutting the dendrogram at a specific height may yield results that are characterised by low values for the quantitative metrics. It can also be argued that, because of the intrinsic properties of résumés and the information contained in them, the decision boundaries between two categories may be not very crisp, and therefore the similarities among résumés of different yet sim-

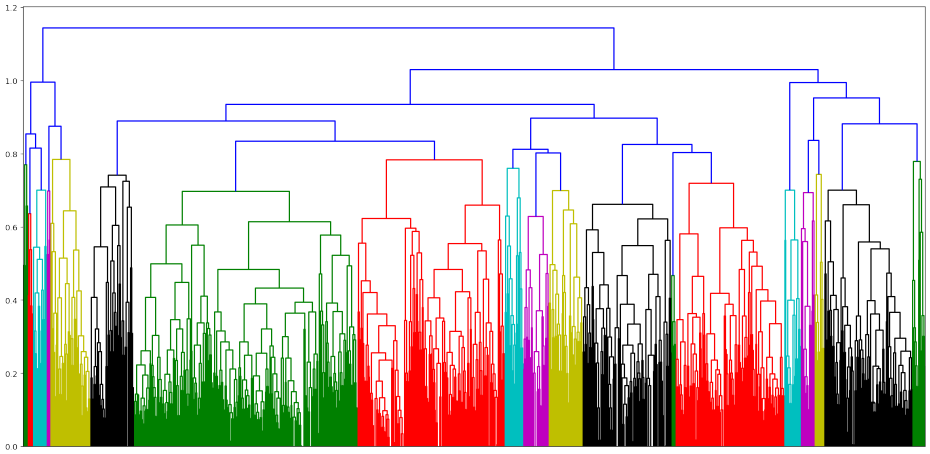


Figure 4.9: Dendrogram with BM25-TextRank-based summarization method.

ilar job profiles, such as for example “Engineering” and “Information Technology”, and “Accounting” and “Finance”, could be rather high, creating denser areas in the feature space.

In order to evaluate this hypothesis, the obtained clusterings were also qualitatively evaluated. In doing so, it is noticeable that, as expected, each clustering actually produces a noisy cluster, containing data objects for several different classes. The impact of such cluster on the overall clustering performances can be evaluated by simply removing it and recomputing metrics. As an example, one of the best clusterings is considered, namely the one where BM25-TextRank is used for summarization, with cut at a height of 0.65. Such clustering distinguish 43 different clusters. Albeit the F1-score of this clustering is lower than others, its ARI is significantly higher.

Table 4.6 shows the metrics before and after the filtering. Removing the noisiest cluster guarantees a significant improvement across all considered metrics. It is possible that the removed cluster actually contains résumés characterised by a set of skills and competences which can represent different categories. It is likely that, at deeper levels of the dendrogram, more specific and cohesive clusters are characterised by predominant categories. In order to verify this intuition, the clusters in the underlying level of the dendrogram are analyzed. In Fig. 4.10, the noisy cluster and the its two child clusters are shown, namely those clusters which had been merged into the noise one. It can be observed that these clusters present a distribution of the categories in which two categories are highly prominent with respect to the others. This effect is still more evident in the clusters derived from these clusters.

Thus, the availability of the dendrogram is particularly useful when dealing with the assignment of a profile to an unseen résumé. The minimum distance between a résumé in the dendrogram and the unseen résumé can be considered. Then we can track where the closest résumé is placed in a cluster at each level of the dendrogram

Table 4.3: Best clustering results with no summarization in terms of BCubed F1-score

Distance	Clusters	Precision	Recall	F1-Score	ARI
0.55	35	0.30	0.26	0.28	0.163
0.50	55	0.35	0.22	0.27	0.155
0.60	23	0.24	0.31	0.27	0.131
0.65	18	0.21	0.34	0.26	0.100

Table 4.4: Best clustering results with k-means summarization in terms of BCubed F1-score

Distance	Clusters	Precision	Recall	F1-Score	ARI
0.60	41	0.15	0.27	0.20	0.047
0.55	56	0.19	0.21	0.20	0.055
0.50	71	0.21	0.17	0.19	0.067
0.65	27	0.13	0.35	0.19	0.039

Table 4.5: Best clustering results with BM25-TextRank in terms of BCubed F1-score

Distance	Clusters	Precision	Recall	F1-Score	ARI
0.80	19	0.28	0.45	0.35	0.195
0.75	24	0.29	0.43	0.35	0.196
0.70	33	0.32	0.40	0.35	0.198
0.65	43	0.36	0.32	0.34	0.222

Clustering	Precision	Recall	F1-score	ARI
BM25-TextRank-Unfiltered	0.36	0.32	0.34	0.22
BM25-TextRank-Filtered	0.41	0.34	0.37	0.28

Table 4.6: Results for BCubed Precision, Recall, F1-score, and ARI for the clustering performed with BM25-TextRank, for the whole clustering (-Unfiltered), and after filtering the noise cluster (-Filtered)

until we determine a cluster characterised with keywords that identify a profile: these keywords can be identified by using *word clouds*. Actually, words highlighted in the word clouds characterise precisely the profile corresponding to the cluster. To verify this observation, each obtained cluster was explored by plotting the distribution of categories and word clouds generated from the résumés in the cluster. For the sake of brevity, only some clusters are reported in Figure 4.11.

The analysis of the clusters allows for several interesting observations. First, it is noticeable that in all clusters, at most three categories are highly represented, whereas the others have a rather low incidence, as for example shown in Figure

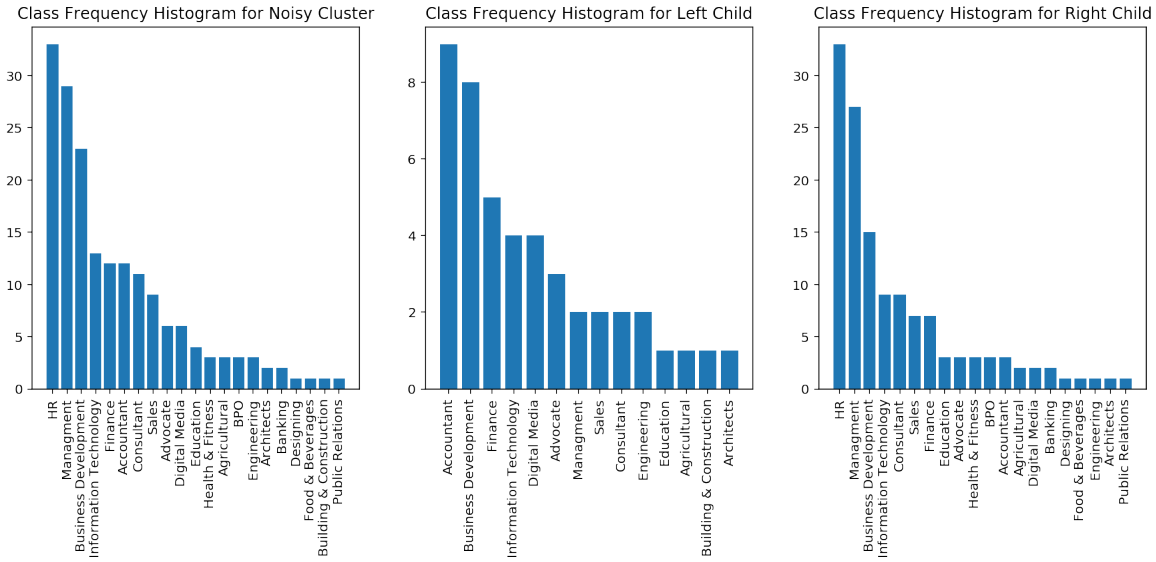


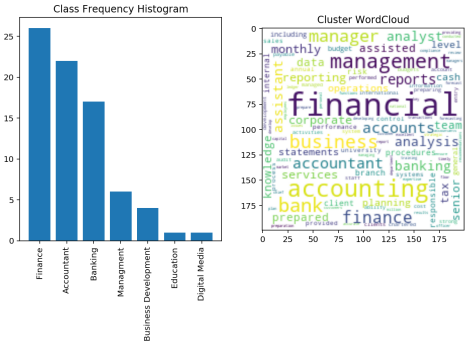
Figure 4.10: Distribution of categories in the noisy cluster and its left and right child.

4.11a. Second, in several clusters such as for example those in Figures 4.11d and 4.11e, only one category has a high incidence in the cluster, while the others are instead underrepresented in terms of number of résumés in the cluster. Finally, it can be argued that, in the case that two or more categories have a high frequency in the cluster, such categories are very similar to one another. "Finance", "Accountant" and "Banking" in Figure 4.11a, and "Engineering" and "Information technology" in Figure 4.11c are clear examples. Moreover, such categories are described by frequent keywords in the word clouds which strongly characterise them with respect to others. This is consistent with the idea that specific job profiles may share knowledge in certain fields, in addition to hard skills. On the other hand, the dataset has been generated by associating a unique category with each résumé, but it is highly likely that each worker can be suitable for different, but similar jobs in the same company. The analysis of the word clouds and in particular of the most frequent words allows assigning a profile to each cluster, which is expected to become even more specialized as the distance threshold decreases, i.e. travelling downwards in the dendrogram.

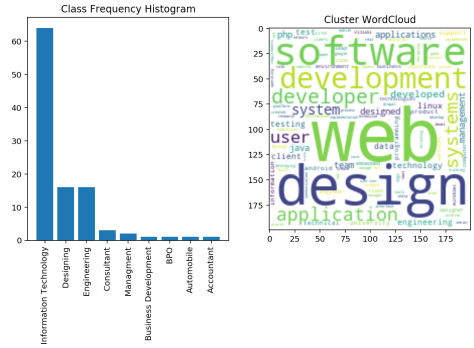
4.3 Discussion and further improvements

In the light of the results obtained by the two proposed approaches, several observations can be made.

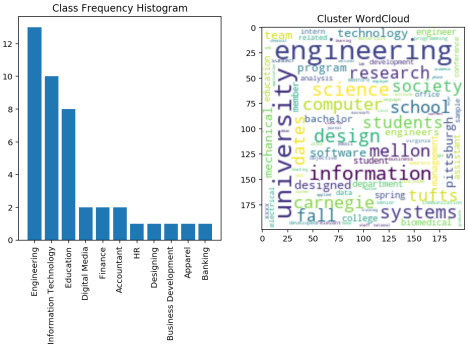
First, unsupervised methods appear to be promising in the specific context. Albeit the values of the quantitative metrics are not excellent, in fact, it is clear that approaching this problem with unsupervised techniques has several advantages.



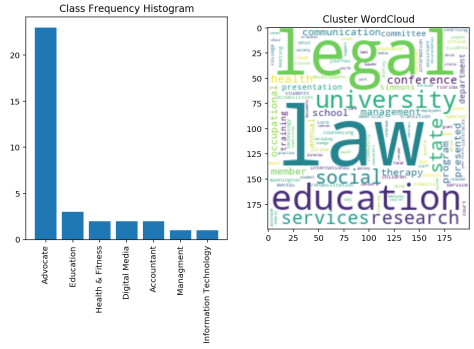
(a) Cluster 40



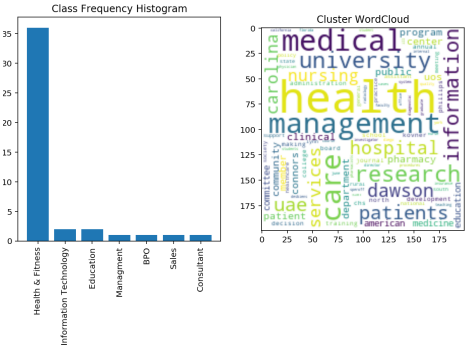
(b) Cluster 31



(c) Cluster 27



(d) Cluster 24



(e) Cluster 10

Figure 4.11: Samples of clusters and categories in each cluster.

- It allows modelling similarities of documents, and thus identifying potential profiles, without prior knowledge or assumptions on the contents of the résumés.

- It alleviates the problem of working with labelled data, that are especially scarce in this domain. To the best of our knowledge, no benchmark dataset exists for résumé profiling or classification. This may be mainly due to two reasons: i) privacy regulations must be addressed when collecting and labelling datasets that, for their scope, must include personal information; ii) labelling such data is rather hard, even for experienced recruiters. Indeed, it is particularly interesting to note that, while the annotation of a résumé with a single profile label is a hard task by itself, since a résumé can describe diverse skills and professional figures, it is rendered harder by the fact that the job market is in continuous evolution. Therefore, profiles may be rather sensitive to concept drift, as new skills and entire profiles could emerge across the years.

Second, all such issues with the construction of the dataset, both in terms of data retrieval and labelling, may affect performances on systems that are based on learning with domain-specific data. This is clearly true in the case of supervised approaches, for which gold labelled data are needed. Such data, even if existing, may not be representative of even slightly different domains, and of how job categories and profiles, and thus labels, evolve over time. However, it may be also true for systems that learn unsupervised models on specific data, such as the proposed model based on keyword extraction and Doc2Vec. As the data evolve over time, the model should be updated or re-trained from scratch in order to allow for the accurate modelling of categories. On the contrary, systems that rely only on purely pre-trained models, and document-wise analysis, may be less prone to such drawbacks and allow for consistent performances over time and on different domains.

Third, summarization appears to be the best method for constructing smaller, more focused versions of résumés, or more generally of documents, with respect to keyword extraction. On the one hand, summarization allows keeping the syntax of entire sentences intact, that can prove to be beneficial when modelling them for tasks of semantic similarity with Transformer-based model, which usually expects whole sentences as input. On the other hand, exploiting a keywords-only approach may hinder the final performances due to the fact that several important keywords or sentences may be discarded from the analysis. In addition, it has been shown how BERT-based methods for summarization underperform in this task. This can serve as further proof that pre-trained BERT-based representation of sentences without further specific fine-tuning are not able to model the semantics of the whole sentences. On the contrary, additional fine tuning such as the one performed by Reimers and Gurevych (2019) enables such semantic modelling of entire sentences. Finally,

it would be interesting as a future work to exploit abstractive summarization. Abstractive summarization is typically performed with Transformer architectures. In this case, the goal is, given a document, to generate a summary from scratch, without necessarily keeping the original sentences. As abstractive summarization has received a lot of attention in the last few years, it would be interesting to evaluate how it can perform on semi-structured texts such as résumés.

Fourth, and most important in the present context, it was shown how Sentence-BERT can be successfully exploited to provide sentence embeddings, and that the computation of whole document embedding via mean pooling is an efficient and effective strategy to model its semantics. Interestingly, such performances can be obtained with pre-trained models with no further learning. As a future direction, it would be interesting to perform additional fine-tuning to such models to adapt them to the specific context, by for example performing additional steps of unsupervised pre-training with domain-specific data, in order to better model the meaning of words in specific contexts.

4.4 Summary

This chapter has presented two approaches aimed at profiling résumés based on their content. Both approaches uses unsupervised techniques to obtain a distributed representation of résumés, and perform hierarchical agglomerative clustering to distinguish profiles.

Section 4.1 has presented the first approach, which employs a keyword extraction algorithm based on Mutual Information and Word Entropy to identify the most relevant keywords for each résumé in a given time span. Such keywords are further used as a representation of each résumé to train a Doc2Vec language model. Thus, distributed representation of résumés based on their keywords are obtained. Finally, an evaluation of the clustering is performed.

In Section 4.2, two techniques for summarization are employed to obtain a shorter, more focused version of the résumé. Then, the résumés thus obtained are fed to a Sentence-BERT model in order to obtain their distributed representation with a pre-trained Transformer-model. Again, clustering is applied to the résumés embeddings. Clustering is evaluated both quantitatively and qualitatively.

Finally, Section 4.3 has discussed the obtained results of both methods and advantages and drawbacks of each one. One important observation is that Sentence-transformer models appear to be the best choice to represent entire documents in a semantically meaningful way.

Chapter 5

Exploiting fact-checking and semantic similarity to perform fake news detection

Fake news are still debated in literature as we discussed in Section 2.2 since they encompass a wide variety of sub-types (e.g. rumours, fact-checking etc.) and impact on several issues such as information reliability, information diffusion and social interactions. The scientific literature reflects this complexity, resulting rather fragmented and leaving important questions still open.

While several definitions have been proposed for rumours and fake news, they arguably share some underlying properties. First, they refer to information that is either potentially or verifiably false. Second, while the intent of the authors and propagators may vary (e.g. humour/satire, misinformation and overt deception), it is clear that the most dangerous fake news and rumours that are spread are those which have a deceptive intent. Third, their most important mean of propagation is social media (Bondielli and Marcelloni, 2019). In the present work, we adopt this broader definition of fake news that takes into account these underlying properties.

Another interesting aspect to be taken into account is how fake news relate to proper news, i.e. real and verified ones. This aspect has not been widely considered by the literature, but it may arguably represent one of the keys to fight the surge of fake news on social media. It is in fact often the case that fake news and rumours are spread in conjunction with events that have a certain relevance to the public opinion, such as newsworthy events, as for example proposed in (Zubiaga et al., 2018a). In this case, fake news and rumours are often spread to mislead users of social media platforms into thinking that the real-world event in question has unfolded differently from what reliable news propose. This is clearly dangerous, both because users are misinformed with respect to important events that happen around the world, and because such misinformation can be used to drive the narrative of

the events in order to steer the public opinion towards certain conclusions. For example, during the Notre Dame fire of April 2019, several fake news emerged, most of them based on the central theme that either the Muslim community or the yellow jackets movement was to blame for starting the fire.

Most of the research concerning fake news and rumour detection treated it as a standard two-class or multi-class classification problem. Content-based or context-based features are typically used to learn supervised models for classification on datasets containing fake and real news or rumorous and non rumorous social media posts (Bondielli and Marcelloni, 2019). In this context, it is important to point out the key limitations of such an approach. First, as the characteristics of breaking fake news or rumours are heavily dependent on the breaking news story that they follow, training such models on limited datasets may make them less able to generalize on the problem, and subject to concept drift issues. Second, classification-based approaches do not take into account an important aspect of fake news and rumours, that is how to actually verify the information contained in them. Once trained on relevant data, classifiers can predict with reasonable accuracy if a piece of news on a specific topic represents a real or fake news, or a rumour. However, arguably they cannot answer the question about the reasons they are real or fake, such as supporting or denying statements from reliable news channels for example. This kind of information could nonetheless prove to be invaluable both from a research perspective and a user-centered one.

One possible answer to such problem could be to address the task of fake news and rumour detection while taking into account also methods for performing fact-checking. In the last few years several fact-checking initiatives from various actors including journalists, governments, organizations, and companies have been encouraged. In the past, fact-checking was typically performed manually, resulting in the collection of large amounts of annotated resources for this specific task. More recently, researchers have started to use such resources with the aim of training automatic fact-checking systems (Popat et al., 2017a; Shaar et al., 2020a; Wang, 2017). The task of computational-oriented fact-checking is undoubtedly hard to address, but it may nonetheless pose a fundamental building block in the construction of automatic systems for fighting misinformation. Typically, the goal of computational-oriented fact-checking is to develop systems able to fact-check a given statement based on provided information, that is to find evidence in support of the given statement in a knowledge base that represents factual information of some sort.

Arguably, fake news detection configures itself as the opposite task of fact-checking. If for fact-checking the goal is to identify information that is actually supported by facts, fake news are the exact opposite, i.e. news that are not supported by factual information. Thus, when considering a system for fake news detection that can move further from classification techniques based on surface properties of texts

or diffusion patterns, it may be interesting to look into properties of fact-checking systems, and incorporate their notions in a fake news detection system. Specifically, rather than proving something is actually false, it could be viable to perform the task by identifying information that is supported by facts, and consider other candidates as potentially fake. The term potentially here plays an important role. In fact, while a system developed with these idea in mind may not obtain the same performances of a simpler classifier in a constrained environment, it may prove to be much more helpful in real case scenarios, where the information flow is continuously modified and where users actually need guidance in determining what to trust. From a user-focused perspective in fact, it may suffice to propose systems that, rather than strictly classifying pieces of news as real or fake, trigger warnings in the case that no available factual information supports a given claim, in order to empower users with the possibility of evaluating potentially misleading news.

From a technical standpoint, this goal could be achieved by developing fact-checking systems that can be trained, either supervisedly or unsupervisedly, to recognize pair of claims that verify each other with a certain degree of confidence. Then, given the assumption that verified claims or news are available, the system could be incorporated in a fake news detection system in order to trigger warnings when little to no factual information is found to verify a piece of news, that could in turn represent a fake news. This could be achieved by focusing on several underlying properties that describe fake news highlighted above. First, the fact that they resemble proper fake news, and thus, without any knowledge of facts, can be easily misinterpreted as real news. Second, the fact that despite the absence of verifiable information, they are often spread in conjunction with important breaking news events, by modifying some crucial aspects of their narrative in order to mislead readers. Thus, in this context taking into account the actual semantics of news may provide an invaluable advantage for the detection. Modelling news in order to encode their meaning rather than surface properties, and determining their veracity by looking at them in comparison with statements that describe facts, may pave the way to more viable strategies to effectively assess fake news that have been confirmed as false, and increase awareness of unconfirmed information for the end users.

The goal of this Chapter is threefold. First, a fact-checking system based on sentence similarity models is proposed. The proposed system has obtained rather promising results in an international evaluation campaign on fact-checking, and specifically on the retrieval of verified claims for unverified ones. The system serves as a case study for evaluating the idea that semantically relevant representation of news can actually be exploited for identifying already verified statements. Second, a methodology for collecting and labelling potentially fake, and certainly verified news from social media for a given event is proposed, in order to obtain real world

datasets that include both real and fake news regarding specific events, thus enabling more real-world focused analysis. Third, the proposed fact-checking system is adapted to perform fake news detection, and experimented on real world real and fake news in order to evaluate its capabilities in identifying potentially fake claims.

5.1 Fact-checking with sentence similarity

Earliest work on automated fact-checking defined the task as *the assignment of a truth value to a claim made in a particular context* (Vlachos and Riedel, 2014). Most approaches on automated fact-checking exploit the reliability of a source and the stance of its claims with respect to other claims and already verified information. The assignment of the truth value is often based on the way in which particular claims (or rumors) are spread on social media (Canini et al., 2011; Castillo et al., 2011; Gorrell et al., 2019; Shu et al., 2017) or on the Web (Mukherjee and Weikum, 2015; Popat et al., 2017b). Other approaches use Wikipedia (Nie et al., 2019; Thorne et al., 2018) or other knowledge graphs (Ciampaglia et al., 2015; Shiralkar et al., 2017) to fact-check claims. More recently, a novel approach has been proposed that exploits Sentence-BERT (Reimers and Gurevych, 2019) to re-rank claims (Shaar et al., 2020a) in order to predict whether a claim has been fact-checked before. Systems able to decide whether a claim has been already fact-checked have become particularly relevant, because they contribute to breaking down the costs of verifying both old and new viral claims.

The task is indeed strongly related to the concepts of information extraction and text similarity. Thus, in order to tackle the challenge of automated fact-checking, it is possible to start from two main assumptions. Intuitively, to decide if two claims are related to each other, it is important to establish whether i.) they share some linguistic properties (e.g., mentioned entities) and ii.) they are in general semantically similar. In order to deal with i.), traditional Information Extraction (IE) methods are still very relevant and accurate (Qi et al., 2020) when extracting information such as Named Entities (e.g., persons, locations and organizations) and content words (e.g., nouns, verbs). On the other hand, ii.) requires a deeper representation of the meaning of the entire text. As already claimed and demonstrated in the previous Chapters, such representation can be obtained with Neural Language Models (Bengio et al., 2003). State-of-the-art Language Models such as BERT and GPT (Devlin et al., 2019; Radford, 2018) based on *Transformer* architectures and *attention mechanisms* (Vaswani et al., 2017) have in fact become increasingly popular in the last couple of years, thanks to their ability to model whole text sequences and generate pre-trained representations that can be fine-tuned for different tasks. Specifically, Sentence-BERT models have proven their effectiveness in modelling the semantic similarity of sequences of text (Reimers and Gurevych, 2019).

The proposed approach to fact-checking is modelled on such assumptions, and has been originally developed by Passaro et al. (2020) in order to face the task "Verified Claim Retrieval" (Task 2) for the CheckThat! 2020 evaluation campaign (Arampatzis et al., 2020; Barrón-Cedeño et al., 2020; Cappellato et al., 2020; Shaar et al., 2020b). The task has been organized with the goal of supporting journalists and fact-checkers when trying to determine whether a claim has been already fact-checked. The goal of the task was specified as follows: "Given a check-worthy claim and a dataset of verified claims, rank the verified claims, so that those that verify the input claim (or a sub-claim in it) are ranked on top".[1]

Thus, given a tweet (the check-worthy claim) and a set of already verified claims (*vclaims*), the goal of the task is to predict, for every *target tweet-vclaim pair*, the likelihood of the vclaim verifying the tweet. Indeed, among the target tweet-vclaim pairs, there exists only a *gold pair* whose vclaim verifies the tweet, which therefore is a correct match. The goal is achieved by ranking, for each tweet, the claims that are more likely to verify it. The dataset is composed of three elements:

1. the verified claims used for fact-checking, each of them provided with an identifier, a title, and the actual claim;

2. the training tweets, associated with an identifier and a textual content;

3. the correct pairing between tweets and verified claims.

The training set provided by the task organizers consists of 1,003 tweets (803 for training, 200 for development) and 10,373 already verified claims. The test set consists of 200 additional tweets.

The system is based on the two previously mentioned assumptions: the claims that verify a tweet are expected to mention the same entities and keyphrases and should have a similar meaning. To address the first point, among target tweet-vclaim pairs, a subset of *candidate pairs* (also referred as *potential pairs*) is identified in which the tweet and the vclaim share at least a named entity or a content word. We refer to the step of identifying candidate pairs among target ones as the *IE step*. Subsequently, in order to estimate the text similarity between a tweet and a vclaim, Sentence-BERT (Reimers and Gurevych, 2019) is exploited to create a language model that is able to better deal with sentence-level textual similarity. This model is then used to learn if a claim can be used to verify a tweet. In particular, two cascade fine-tuning steps are performed, aimed at i.) assigning a higher cosine similarity to gold tweet-vclaim pairs and ii.) actually classifying a target tweet-vclaim pair, and more specifically a candidate tweet-vclaim pair, as a correct match (gold) or not. First, the Sentence-BERT model is fine-tuned following the same paradigm

[1] `https://github.com/sshaar/clef2020-factchecking-task2`

proposed in Reimers and Gurevych (2019) in order to assign a higher cosine similarity to gold tweet-vclaim pairs. Then the resulting model is further fine-tuned to decide, given a candidate tweet-vclaim pair, whether the tweet is actually verified by that claim or not. This is achieved by training a sentence-pair classifier to label each tweet-vclaim pair as a correct (gold) match or not.

Information Extraction module

Starting from the assumption that similar claims tend to mention the same entities and keyphrases, an IE module was developed to find potential tweet-vclaim pairs. The module is based on Stanza (Qi et al., 2020), a state of the art natural language analysis package. Each text fragment (i.e., a tweet, a vclaim or a vclaim title) was processsed by applying Sentence Splitting, PoS-tagging, Lemmatization, and Named Entity Recognition. Thus, each text is associated with its keywords, consisting of its content words (nouns, verbs, and adjectives) and named entities.

vclaim and title	**keywords**
title: Was Sen. Chuck Schumer a Client of 'Hollywood Madam' Heidi Fleiss? **vclaim:** Sen. Chuck Schumer's name and/or phone number were found in "Hollywood Madam" Heidi Fleiss's black book of clients.	'chuck schumer', 'hollywood', 'heidi fleiss's', 'chuck schumer', 'hollywood', 'heidi fleiss', 'sen.', 'chuck', 'schumer', 'phone', 'number', 'find', 'hollywood', 'madam', 'heidi', 'fleiss', 'black', 'book', 'client', 'sen.', 'chuck', 'schumer', 'client', 'hollywood', 'madam', 'heidi', 'fleiss'

Table 5.1: Example of relevant lemmas and named entities in a claim.

tweet	**keywords**
Chuck Schumer was one of Hedil Fleiss' top clients. Look it up. Doug Masters (@protestertrophy) January 23, 2019	['chuck schumer', 'hedil fleiss', 'doug masters', 'january 23, 2019', 'chuck', 'schumer', 'hedil', 'fleiss', 'client', 'look', 'doug', 'masters', '@protestertrophy', 'january']

Table 5.2: Example of relevant lemmas and named entities in a tweet.

Given a tweet, in order to retrieve potential claims that verify it, we used two different functions based on the keywords:

IE function – the overlapping score is simply computed by counting the number of elements (cf. the keywords field in Table 5.1 and Table 5.2) shared by the tweet and the claim. Candidate tweet-vclaim pairs are required to share at least one lowercased element (named entity or content word).

IEElastic function – it exploits Elasticsearch[2] to find the potential candidate pairs. Specifically, for each tweet, candidate claims consist of the top 1,000 matches ranked by relevance, using the scoring function provided by the task organizers for the baseline. Such scoring function is an Elasticsearch multi-match query based on both the vclaim and its title and the tweet itself.

Candidate tweet-claim pairs obtained with the IE overlapping function were used to obtain a training set for the first fine-tuning of the transformer model. Candidate tweet-claim pairs obtained with the IE and IEElastic functions have been also used at inference time to obtain the final predictions submitted for evaluation.

Cascade fine-tuning of Transformer models

Sentence-BERT models have proven their effectiveness in tasks related so Semantic Textual Similarity (STS) (Cer et al., 2017). In order to train the system to better recognize gold tweet-vclaim pairs, their semantic similarity between tweets and claims belonging to the same candidate tweet-vclaim pairs is taken into account. This is achieved by further training an already fine-tuned Sentence-BERT model, namely `bert-base-nli-mean-tokens` (Reimers and Gurevych, 2019), available within the sentence-transformers Python library.[3]

The `bert-base-nli-mean-tokens` model was originally trained on SNLI (Bowman et al., 2015) and MultiNLI dataset (Williams et al., 2018) and tested on the STSbenchmark (Cer et al., 2017). During the original training, which can be seen as a fine tuning of a pre-trained BERT model (Devlin et al., 2019), a classifier is tasked to annotate pairs of sentences from SNLI and MultiNLI with the labels *entail*, *contradict*, and *neutral*. The evaluation was performed on the STSbenchmark (Cer et al., 2017) dataset, which contains sentence pairs and their similarity score. The trained model was exploited to infer sentence pair similarity via cosine. The `bert-base-nli-mean-tokens` achieved 77.12 Pearson correlation with gold scores on the STSbenchmark test set.

Starting from the `bert-base-nli-mean-tokens` model, two levels of fine-tuning are added in order to adapt the sentence pair similarity task to the fact-checking one (i.e., gold tweet-vclaim pairs are associated with the maximum cosine similarity), and to fact-check a pair with a classification layer (i.e., gold tweet-vclaim pairs are associated with the positive label). In order to train the model, both the verified claim and its title are used as examples. The usage of both the vclaim and vclaim_title for training has two main advantages. First, it allows to increase the size of the dataset so that the model can be learned by using a higher number of positive examples, that are under-represented. Second, it helps adding variability to the

[2] `https://www.elastic.co/`

[3] `https://github.com/UKPLab/sentence-transformers`

training examples, both positive and negative. For example, a title may contain an acronym such as "KKK", whereas the claim may contain its extended form, in this case "Ku Klux Klan". In our experiments we noticed that such a variability was very helpful to improve the overall performances of our models.

Sentence pair similarity of tweets and vclaims

The first fine tuning step consists in training the model to output sentence embeddings for tweets and vclaims that are closer to each other in the n-dimensional space for gold tweet-vclaim pairs.

In order to do so, the fine-tuning proposed in Reimers and Gurevych (2019) was followed. Specifically, authors used the STSbenchmark (Cer et al., 2017) dataset, containing pairs of sentences with a similarity score ranging from 0 (no similarity) to 5 (maximum similarity). The Sentence-BERT model was fine-tuned using the regression objective function on the training set. Therefore, for each epoch, loss was computed by considering the correlation between the gold similarity judgments and the predicted cosine similarity between sentence embeddings.

In the case of fact-checking, the model is trained to assign the highest possible cosine similarity to gold tweet-vclaim pairs, thus separating them from other candidates. Given the assumption that a claim that verifies a tweet is semantically similar to it, the training set was built as follows:

1. two positive examples were created from a gold tweet-vclaim pair, the first one composed by the tweet and the vclaim itself (tweet-vclaim pair), and the second one composed by the tweet and the title of the claim (tweet-vclaim_-title). Both the positive pairs were assigned with a cosine similarity value of 1.0. This forces the model to boost the similarity between the texts belonging to gold pairs;

2. for each gold tweet-vclaim pair, 20 other tweet-vclaim pairs were randomly selected as negative examples from the list of candidate pairs obtained with the IE overlapping function (cf. Section 5.1). The similarity of the negative examples was computed as the cosine similarity between vectors predicted by `bert-base-nli-mean-tokens`, modified by the *tanh* function. This has the effect of decreasing the cosine similarity, thus effectively penalising negative examples.

The model, named `bert-base-nli-factcheck-cos`, was trained for 4 epochs with a batch size of 8, and 10% of data were used for warm up.

Classification of matching pairs

For the second fine-tuning step, the model is trained on a simple binary classification task to distinguish between matching (gold) pairs, labelled as 1, and non-matching ones, labelled as 0. Similarly to the previous fine-tuning step, negative examples were selected among candidate tweet-vclaim pairs returned by the IE module. Like for the sentence pair similarity model, for each tweet, the tweet-vclaim and the tweet-vclaim_title pairs were used as positive examples. However, in this case 2 negative examples were selected among the tweet-vclaim candidate pairs, in order to better balance the training data for the classification.

The model, `bert-base-nli-factcheck-clas`, is therefore a Transformer with a classification head on top of it, implemented with the Huggingface library.[4] The model was initialized with the weights of `bert-base-nli-factcheck-cos`, and was trained for 3 epochs with a batch size of 8. The AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $2e-5$ was used.

Experiments and results

All the experiments were performed on the available dataset provided by the task authors.

Once the models were trained, the `bert-base-nli-factcheck-clas` classification model is used at inference time to classify candidate tweet-vclaim pairs. To retrieve the potential candidates, the IE and IEElastic function described in Section 5.1 were used. However, an experiment using all possible tweet-vclaim pairs was performed as well.

As the goal is to provide a ranking of the claims that are most likely to verify a tweet, in all the cases the probability of class 1 (i.e. the tweet-vclaim pair is a gold pair) outputted by the `bert-base-nli-factcheck-clas` classifier was used rank the vclaims for each tweet.

The evaluation metric used in the competition was the Mean Average Precision@5 (MAP@5) calculated over the gold ranking. Results are reported in Table 5.3. The table shows the performances of each module obtained on the task by ranking the claim for a tweet according to several measures. More specifically, for each module, we report the model name, the type of the fine-tuning we applied, the function used at inference time for selecting candidates and the MAP@5 obtained with the official scorer. As for the IE step, given a tweet, the claims were ranked according to the overlapping function for both the IE and the IEElastic methods. The IEElastic method coincides actually with the baseline provided by the task organizers. To assess the performances of the Sentence-BERT model fine-tuned on cosine similarity, namely the `bert-base-nli-factcheck-cos`, claims were ranked according to

[4] `https://huggingface.co`

the adjusted cosine similarity. As for the final results, at inference time, the model `bert-base-nli-factcheck-clas` was fed with the candidate tweet-vclaim pairs calculated with the IE and the IEElastic methods, and with no candidate pre-selection.

Model	Fine-tuining	Inference	MAP@5
`bert-base-nli-factcheck-clas`	classification	IE	0.916
`bert-base-nli-factcheck-clas`	classification	IEElastic	0.912
`bert-base-nli-factcheck-clas`	classification	-	0.89
IEElastic baseline	-	IEElastic	0.815
IE baseline	-	IE	0.74
`bert-base-nli-factcheck-cos`	cosine similarity	IE	0.41
`bert-base-nli-factcheck-cos`	cosine similarity	IEElastic	0.35

Table 5.3: Results calculated for each module of the architecture.

The proposed system was submitted to the Task 2 for the CheckThat! 2020 evaluation campaign (Arampatzis et al., 2020; Barrón-Cedeño et al., 2020; Cappellato et al., 2020; Shaar et al., 2020b) by the UNIPI-NLE team (Passaro et al., 2020). It ranked second among participants. Table 5.4 reports the performances of the team model and those of the winning system and task baseline for reference.

Team	MAP@1	MAP@3	MAP@5
Buster.ai	0.897	0.926	0.929
UNPI-NLE	**0.877**	**0.913**	**0.916**
Task Organizers	0.767	0.812	0.815

Table 5.4: Performance of the UNIPI-NLE models against the top performing model and the official baseline (Passaro et al., 2020).

By analysing the obtained scores, it is clear that the proposed approach is able to outperform the baseline by a wide margin, despite the fact that the Elasticsearch based approach proposed by the task organizer was shown to be very effective nonetheless.

Moreover, several interesting insights can be drawn from the various steps of the proposed approach. The IE baseline appears to be less effective as a standalone tool for selecting the best candidates among claims for each tweet, with results below the IEElastic one. However, the IE method performs optimally when used as a selection criterion at inference time. In addition, when the classifier was shown with all possible tweet-vclaim pairs with no pre-selection, performances degrade noticeably. Arguably, this is due to the fact that the classifier is trained to distinguish between the gold tweet-vclaim pair and other pairs that share similar features but are in fact incorrect. Therefore, the classifier may be more prone to errors when tweet-vclaim pairs, which differ greatly from each other, are shown as it never saw

such examples during training. Experimental results seem to confirm such hypothesis. This could be considered as a potential shortcoming for the classifier itself. Nevertheless, it gives two potential advantages. First, the classifier needs a lower number of negative examples for an effective training. We can argue that it is more difficult to decide between two similar claims for a tweet, rather than between two very different ones. Therefore, we chose to train the classifier to solve the "harder" problem, and addressed the "simpler" one with a less sophisticated, yet effective, approach. Second, the classification of each tweet-vclaim pair is time consuming, and this is expected to become an issue when the system is used at scale and tens of thousands of pairs have to be classified. The IE method is efficient because it only needs to extract content words and named entities for each pair, a task that is almost trivial in terms of time complexity with modern NLP toolkits and current hardware.

Finally, it is interesting to point out the contribution of the cosine similarity adaptation performed with Sentence-BERT. Clearly, the model itself does not perform well on the present task. However, two observations can be made. First, by using a standard BERT model such as BERT-base-uncased for representing sentences (i.e., by averaging word-level representations obtained from the model), ranking claims based on cosine similarity were completely ineffective, obtaining a very low MAP@5. Instead, by exploiting a pre-trained Sentence-BERT model, much more encouraging results were obtained, that were subsequently improved thanks to our cascade fine-tuning strategy. This serves as additional evidence for the fact that standard BERT models are not able to represent sentences in a semantically proper way, as already claimed in the literature (Reimers and Gurevych, 2019). Second, by exploiting the fine-tuned Sentence-BERT model (`bert-base-nli-factcheck-cos`) for obtaining the initial weights for the classifier (`bert-base-nli-factcheck-clas`), it clearly outperformed a model based on BERT-base-uncased and trained in the same way. More specifically, on the development set a 0.72 MAP@5 was obtained for a `bert-base-uncased` fine-tuned model and a MAP@5 of 0.78 for the `bert-base-nli-factcheck-cos`.

Discussion

Aside from the results obtained on the specific task and dataset, several interesting observations can be derived from the proposed approach.

First, it is clear that, once again, the idea of exploiting pre-trained resources for encoding semantics has proven to be rather effective and useful. Specifically, models trained on objectives whose primary goal is to reward semantic similarity of sequences of text have shown the best performances, and appear to be able to actually understand and model the semantics of entire sentences, which may be crucial in a wide array of tasks.

Second, it is also interesting to compare the performance of base BERT models with respect to Sentence-BERT models on the same task. The fine-tuned Sentence-BERT model in fact clearly outperformed the BERT one. This could mean that, when dealing with tasks that actually require a more thorough modelling of the semantics of the entire sequence or sentence, rather than of the single words, models specifically built for that kind of representation may provide more accurate and descriptive features. This is evident both in the case of simply comparing the cosine between sequence vectors, where Sentence-BERT has a clear upper hand with respect to the sentence representations provided by BERT, that seems to encode entirely different information, and when fine-tuning the model to a downstream task such as sentence-pair classification.

Third, the obtained results prove how the proposed approach based on a combination of Information Extraction and Deep Learning strategies can be viable for performing the task of fact-checking. Concerning the Information Extraction, it gives the approach an advantage both in terms of computation time, as a lower number of pairs must be analyzed, and in terms of performances, as extremely different examples that may fool the classifier are disregarded. As for the transformer model, it has proven to be very effective in preforming transfer learning, by fine-tuning large pre-trained models for specific tasks such as the fact-checking one.

One area that remains to be explored is how the model is able to generalize on other, potentially very different, data. This may be crucial in the development of systems that can face the fake news problem in a real world scenario, and without requiring specific knowledge regarding the currently analyzed event. This question is answered in Section 5.3.

5.2 Collection of a real world fake news dataset: the Notre Dame Fire Dataset

Most of the existing methods for fake news and rumour detection, and computational fact-checking, are focused on modelling the problem as a classification task on restricted datasets.

Some efforts in the direction of building large-scale datasets have been made. Several challenges have to be addressed when collecting data for fake news and rumour detection, such as the identification of relevant data, the relative sparsity of it on social media and news websites, and current regulations concerning data access from social media (Bondielli and Marcelloni, 2019). From a fact-checking perspective, the main focus has been on the collection of statements, especially from politicians and public figures, which are usually associated with a truthfulness rating and information regarding the context, and a brief description of the reason behind the

assigned rating or related news articles (Vlachos and Riedel, 2014; Ferreira and Vlachos, 2016; Wang, 2017). Alterations of pre-existing resources such as Wikipedia to obtain the same results have been proposed as well (Thorne et al., 2018). Concerning instead social media, a lot of attention has focused on Twitter, as collecting larger-scale datasets is easier. Most of the works regarding twitter focus specifically on the collection and annotation of data for various rumour-related tasks (e.g., rumour detection, stance classification) and on the credibility of users in the platform (Mitra and Gilbert, 2015; Zubiaga et al., 2016; Zubiaga et al., 2016; Derczynski et al., 2017). One of the most popular repositories of fake news has been proposed in Shu et al. (2017, 2020). The FakeNewsNet dataset incorporates both the content of the fake news and its diffusion patterns.

However, two key issues concerning fake news datasets should be discussed. First, since most approaches focused on classifying real and fake news in isolation, based on their surface or text properties and the shape of their diffusion patterns, the context, i.e. the real news that surround the fake news, is not considered. This scenario is potentially limiting in terms of generalization capabilities, as systems are not tasked to learn when a fake news emerge with respect to the real ones, and thus how to reason regarding fake news, but rather to simply distinguish what characterizes fake from real news in the specific context of the proposed dataset. Thus, these kinds of systems may be more prone to concept drift and less able to generalize on the idea of fake and real news in real world scenarios (Bondielli and Marcelloni, 2019). Second, most such approaches and datasets often disregard the fact that fake news have different levels of deceitfulness. Some fake news may be rather easy to identify, even exploiting only surface properties, as proposed in the earliest computational approaches to detection (Castillo et al., 2011; Zhang et al., 2012; Pérez-Rosas and Mihalcea, 2015; Rubin et al., 2015), because they typically contain telltale shallow textual and linguistic clues (e.g., hyperboles, repeated punctuation, etc.), and they refer to totally made-up, implausible events. However, as the goal of fake news to fool the readers, they closely imitate the style and language of real ones, and are often based on false but plausible events concerning public figures or real world events, thereby requiring deeper text understanding abilities and a richer knowledge of the actual facts to be identified. In other words, in addition to comprehension skills, the process of fake news detection may benefit from verification of all or some of the facts reported in the news. Therefore, a methodology to collect and label datasets that are better suited to represent real-world scenarios is proposed. A real-world scenario in this case contains, for a specific real-world event, the following information: i) fake news, both easy and hard to detect ones; ii) real news, for example shared by users on social media from news websites; iii) contextual information, intended as a ground truth for the event and its simultaneous events. Datasets collected as such would be able to better describe the interplay

between real and fake news in a given time span, bringing to light also fake news that are harder to discover via surface properties, which arguably represent a major challenge in this field.

In order to collect and label a dataset that can include both real and fake news, and their context for a specific event, the proposed methodology leverages a top-down collection strategy (Zubiaga et al., 2018a), to ensure that collected data contain at least some previously known fake news, and a multi-step crowdsourcing annotation to obtain real and fake labels that can also better reflect the complexity of specific fake news texts.

Collection strategy

As for the collection strategy, the top-down approach was preferred, as the focus is towards specific events or public figures, the approach requires a-priori knowledge of fake news that emerged in connection to the event or the public figure. Moreover, as stated in Zubiaga et al. (2018a), a bottom-up strategy relies on the expertise of annotators in order to label the dataset. However, including experts in the annotation process is both expensive and time consuming.

A viable strategy to collect the data is to exploit social media posts as the source. The advantage of this approach is twofold. On the one hand, social media have been proven to be the most fertile ground for the spread of fake or unverified information (Zubiaga et al., 2018a). On the other hand, social media posts that spread fake news often include links to the actual article which contains the false information. Therefore, from the same collection of data, two different kinds of information, namely the posts and the articles, can be obtained. As for the actual implementation, Twitter was chosen as the social media of reference because it is one of the most widely studied social networks, especially considering rumours and fake news, and fewer limitations are imposed to researchers on the quantity and quality of collected information.

In order to ensure the representativeness with respect to a real-case scenario of fake news diffusion, three different types of data are considered for any given event or public figure:

Subject-Trusted: Posts and linked articles on the subject that have been produced by *trusted news sources*. Trusted news sources are a subsets of users in the platform that are very unlikely to share fake news, such as national or international newspapers accounts and news agencies.

Subject-Untrusted: Posts and linked articles on the subject, that have been produced by *all the other users*. These might include verified users that tweeted about the subject but are not among the trusted news sources.

Context-Trusted: Posts and linked articles that do *not* talk about the subject but have been produced by the trusted news sources *during the time-frame of interest*.

Clearly, the most interesting aspect of the data is expected to be the *Subject-Untrusted* elements. These are expected to contain both real news, shared by users outside of the trusted news sources, and fake news. The *Subject-Trusted* and the *Context-Trusted* subsets will instead serve as the ground truth for the subject of analysis itself and for the selected time-frame.

Twitter allows retrieving posts either by user or by keywords. Both retrieval methods are limited in both the rate and the number of tweets collected. In order to collect subject-specific tweets, a set of keywords is chosen, e.g. small phrases or relevant hashtags. Only tweets that include one or more of them are retrieved. As for the trusted users, their feed can be retrieved entirely, provided that privacy settings allow for it.

Annotation process

The labelling process simply consists in providing a set of tweets and articles with a "Real" and "Fake" label. Specifically, two different annotations are proposed in order to i) obtain a gold annotation for fake and real news and ii) provide information on how difficult the fake news is to identify without knowledge. Operationally, this is achieved by performing two crowdsourcing tasks. The two tasks are structured as follows:

Task 1 - Gold Labelling: In the first task participants are asked to label posts and articles as fake or real news. In order to obtain a gold labelling, annotators are provided with a list of known fake news. The list contains a title for the fake news and a brief description of it, and include the reasons for its non-veracity. Annotators are tasked to identify, for each piece of content (tweet or article), if it is sharing or propagating one of the fake news provided in the list. They can also answer by stating that the text is actually propagating a fake news outside of the provided list. Formally, given a text t (article or post) related to a subject s (event or public figure), and a list F_s of known fake news related to s, participants are asked to decide if t belongs in some capacity to F_s, i.e., if it is (or it is not) sharing or propagating a fake news

Task 2 - harder fake news identification: The second task is similar to the first one, with one key difference. In this case, participants are not given any list of fake or real news, and are asked to label the pieces of content as "Real" or "Fake" without any supporting information. By giving annotators no additional context, it is possible to identify the more complex fake news in the dataset, as the agreement of annotators is expected to be lower with respect to task 1. In order

to speed up the annotation process, the texts rated as "Real" with a very high confidence in task 1 (i.e. $> 90\%$) were removed, as they are less interesting to label and the outcome is expected to be the same in both tasks.

The Notre Dame Fire dataset

As a case study for testing the proposed methodology, the Notre Dame fire of April 2019 was chosen as the reference event. This was due to the fact that the event sparked a lot of controversy and fake news on social media, mostly blaming either the yellow vests or Islamic extremists for the fire. Official sources on the other hand claimed that the fire was not set intentionally. Specifically, seven different fake news were identified during the event. Table 5.5 provides an overview of the fake news and of their context.

The data were first collected by searching Twitter for posts that contain one or more relevant keywords (e.g., *#notredame* and *#notredamefire*), or that are produced by trusted news sources. The considered time span goes from the day before the fire, the 14^{th} April, to two weeks after the fire, the 29^{th} April. As for the trusted sources list, we selected several of the most important English news outlets, such as for example ABC, BBC World, CNBC. The code for crawling Twitter posts and news articles was written in Python. Our crawler adheres to the Twitter guidelines for retrieving tweets.[5]

In total, 153,156 tweets and related 65,434 news articles were collected. News articles were scraped from the link provided in the tweets. Around 76% of tweets contain at least one of the provided keywords, while the rest represent the context news produced by trusted sources. Figure 5.1 shows the distribution of tweets related to the Notre Dame fire for each day.

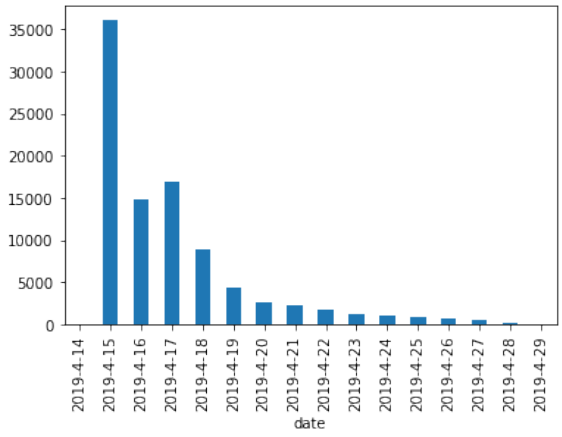


Figure 5.1: Number of tweets related to Notre Dame from 14 to 29 April 2019.

[5]https://twitter.com/robots.txt

	Context	Fake News text
1	A person working on the Notre Dame cathedral at the moment of the fire claims that it was deliberately started and not an accident.	"Notre Dame cathedral in Paris on fire, worker claims it was deliberately started"
2	A fake account for CNN stated that the Notre Dame fire was a terroristic attack.	"CNN can now confirm the Notre Dame fire was caused by an act of terrorism"
3	A tweet claimed that an unmarked car was found in Paris with gas tanks and Arabic documents inside. The news is actually real, but is from 2015, and does not involve Notre Dame.	"Gas tanks and Arabic documents found in unmarked car by Paris"
4	A Muslim girl was allegedly plotting to blow up a car using gas canisters near the Notre Dame cathedral for love.	"Muslim girl in search of love plotted to blow up car packed with gas canisters near Notre Dame Cathedral"
5	A fake Fox News account presented a fabricated tweet from Rep. Ilhan Omar (a Muslim American politician) saying "They reap what they sow" in reference to Notre Dame.	"Ilhan Omar - They reap what they sow #NotreDame"
6	An account tweeted a video of Notre Dame burning with shouts of "Allahu Akbar" edited over the video. Other similar videos were shared after.	"Who yells Allah Ahkbar when they see an 850 yr old beloved and cherished Catholic Religious Monument that has a roaring, blazing fire coming from it?"
7	A low-quality video of a person walking on the outside of the cathedral was used to make false claims about the fire, including that the person shown was an "Imam" or a Yellow Vest protester setting the fire.	"Notre-Dame: who is this person in yellow vest on a tower?"

Table 5.5: Fake news identified for the Notre Dame fire of April 2019.

As for the annotation process, only a subset of the texts was chosen, since labelling the entire dataset with the proposed methodology could be very expensive both in terms of time and economical resources needed for crowdsourcing. However, arguably the subset of gold labelled data could be exploited to bootstrap the annotations for the rest of the dataset, for example by using distant labelling techniques on near duplicates (e.g. retweets, similar news headlines and so on) of the labelled texts. Specifically, the subset contains 573 pieces of content, of which 484 tweets and 89 news articles. The sampling of tweets and articles was chosen to mimic the distribution of tweets and articles in the entire dataset.

Each piece of content, both tweets and news articles, has been labelled by 20 an-

notators for each task. The final True/Fake label is given by the majority vote over labels proposed by the annotators. The agreement between the annotators is computed by means of Fleiss' Kappa (Fleiss, 1971). Fleiss' Kappa can be in fact used to compute the agreement between two or more raters when assigning a categorical label to a number of items. It can be interpreted as the proportion between the observed agreement between raters with respect to the expected agreement if ratings were given at random. The agreement between the annotators, in terms of Fleiss' Kappa, is 0.47 for Task 1 and 0.24 for Task 2. As expected, the agreement on the second task is generally lower as annotators were not provided with any ground truth for the annotation.

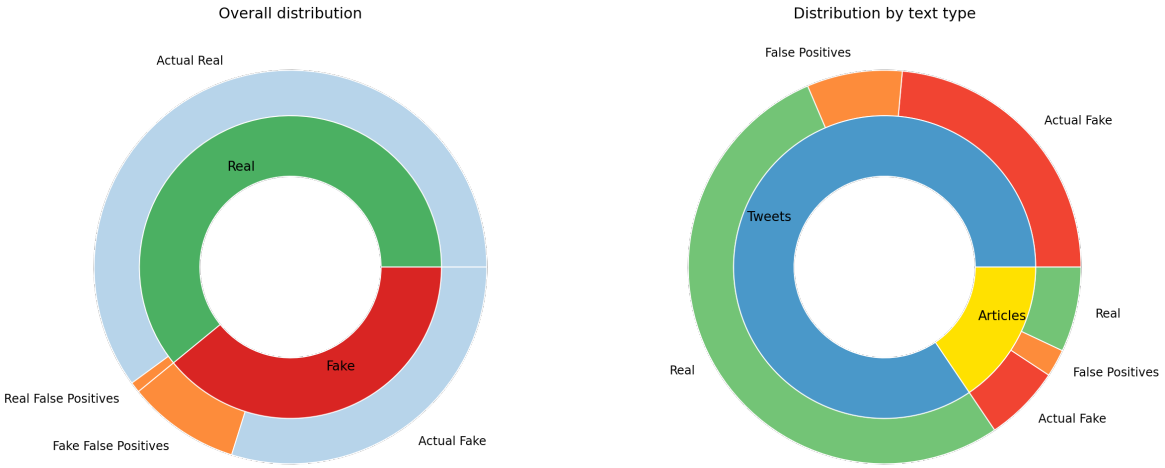


Figure 5.2: Distribution of data and labels for Task 1.

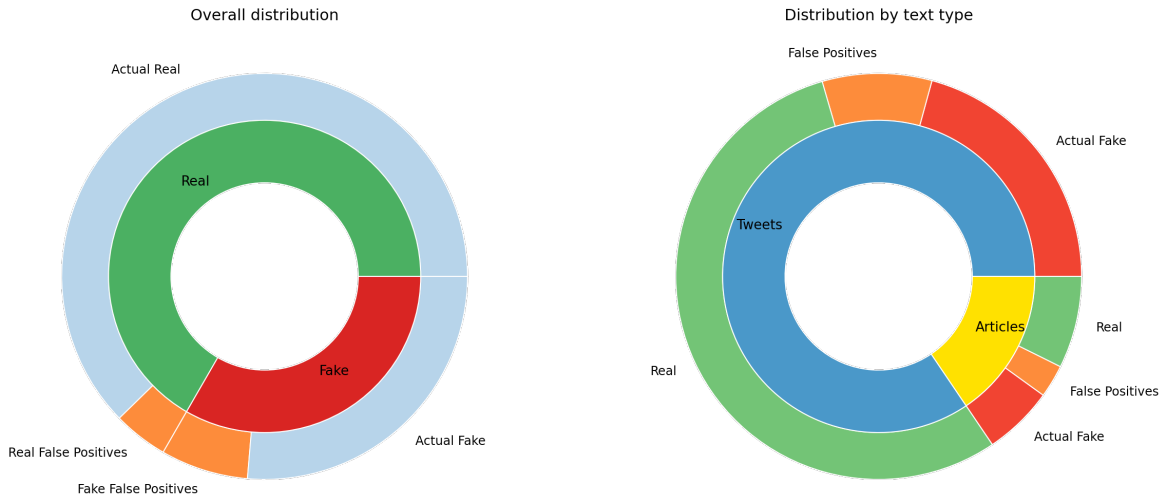


Figure 5.3: Distribution of data and labels for Task 2.

In addition to the labelling via crowdsourcing, each piece of content was manually fact checked in order to ensure the best quality for the dataset. his choice is

twofold. On the one hand, the expert fact-checker's rating can be exploited to determine the majority vote of the news in case of tie (i.e. 10 annotators rated the content as fake, while other 10 rated it as real). On the other hand, we noticed that several news were actually incorrectly labelled in both tasks, probably because of their difficulty in being fact-checked. Specifically, for the first task 58 texts were incorrectly labelled (53 Fake and 5 Real), and for the second task errors were found for 65 texts (40 Fake, 25 Real). It is interesting to notice that, as expected, more misclassifications were observed in the second task. However, while most errors for the first task come from fake news incorrectly classified as real news, the two classes are more balanced for Task 2. The charts in Figures 5.2 and 5.3 show the distribution of data, both in terms of types of text and proportions between real and fake news. In the charts, errors in classification are referred as *false positives*.

5.3 From fact-checking to fake news

The approach for fact-checking proposed in Section 5.1 has unquestionably proved to be promising, both serving as a proving ground for the proposed idea and obtaining rather good performances on the proposed fact-checking task.

Computational-oriented fact-checking has several methodological differences with respect to traditional fake news or rumour detection. However, it is arguably akin to it, especially considering the proposed end goal and the data on which algorithms are applied to. It could be argued in fact that non-verifiable statements may be considered as a sort of fake news, while verifiable ones are what can be defined as a real piece of news. In addition to this, in this context it is also interesting to consider the role of the check-worthiness of claims. It is often the case that claims that are worth to be verified often come from public figures and from news, that is actually the most fertile ground for fake news and disinformation. It is clear how these two aspects are strictly related to each other, as they can be considered two slightly different perspectives on the whole problem of disinformation and misinformation.

Therefore, when keeping in mind these aspects, it would be clearly interesting to attempt at bridging the gap between fake news detection and fact-checking also from a research standpoint. The main research question dwells on how it would be possible to incorporate methods proposed for fact-checking in architectures aimed at fake news detection. From an application viewpoint, it would be interesting to adapt fact-checking systems to work with real and fake news.

Given a ground truth on a specific topic of interest, it can be argued that pieces of news regarding the topic can be actually verified, and thus considered as trustworthy, if evidence of such information is found in the ground truth. Conversely, if little or no evidence is found for a given claim or article, and thus the content is harder to verify given the available ground truth, it may be considered as suspicious and

potentially fake. It is clear that such assumption, and the resulting system, deviates from providing a clear classification between real and fake pieces of news. Rather, it aims to provide a tool that can identify pieces of content or news containing information that is not found in a ground truth. This content may thus be fake and may need a more thorough analysis to determine its veracity. In a real-world application, such method may provide a way to trigger warnings when a piece of news is identified that contains information not verified by official sources, and thus more suspicious.

In this Section, an experiment is proposed to adapt the approach described in Section 5.1 to the task of fake news detection. The approach is evaluated on the Notre Dame Fire dataset (see Section 5.2), that contains real and fake news in the form of tweets and news articles concerning the fire of Notre Dame.

Method

The main goal of the proposed approach is to evaluate how a fact-checking system would perform when applied to the problem of fake news detection. In order to do so, the paradigm of the approach presented in Section 5.1 and in Passaro et al. (2020), is slightly modified to take into account the dichotomy between real and fake news. The original approach was aimed at retrieving verified claims for unverified pieces of content. Specifically, the goal was to rank, for a given set of tweets, a group of verified claims so that the claim that verifies the tweet is ranked on the top (Arampatzis et al., 2020; Barrón-Cedeño et al., 2020; Cappellato et al., 2020; Shaar et al., 2020b).

The proposed method for fact-checking is thoroughly described in Section 5.1. It is based on the idea that claims, or more in general text sequences that can verify other sequences are semantically similar to them. Thus, a pre-trained Sentence-Transformer model (Reimers and Gurevych, 2019) was trained with two cascade fine-tuning steps in order to (i) provide sentence-level features that actually represented the similarity of texts, and (ii) to classify pairs into matching or non matching ones. For each element to verify, a subset of the provided verified claims, obtained with an Information Extraction function, is ranked based on their likelihood of the pair being a correct match.

In the context of fake news, some aspects of the proposed paradigm must be shifted in order to address the higher complexity and different end goal of the task. Let the *ground truth* be a set of texts (e.g. news articles and tweets) that contain verified and fact-checked information: the goal is to identify the real news, that are most likely containing similar information to that included in the ground truth, and thus are *verified* by it, and fake news for which this kind of information is harder to retrieve. Thus, in this case, the proposed solution is as follows.

Model selection. It must be noted that for this experiment, the model provided in Passaro et al. (2020) and described in Section 5.1 was not fine-tuned on the fake news dataset. This is done for (i) better simulating a real-world scenario where labelled datasets for fine tuning are not available, and for (ii) allowing a better understanding of the generalization capabilities of the model in different scenarios and for a rather different task.

Ground truth news ranking. For each piece of news to be verified, the model is used to rank ground truth elements. In Passaro et al. (2020), the ranking was provided based on the probability of class 1 (i.e. a matching pair of texts) outputted by the sentence-pair classifier. In this case the ranking is instead given by considering the probability value for the most likely class for each prediction of the model. In other words, we want to rank higher the predictions for which the model is more confident. This choice is motivated by the fact that in this case we are interested in both ground truth texts that supposedly verify the piece of news, and in texts that are instead very different, and are more likely to provide contrasting information. By exploiting the maximum probability, it is possible to consider as relevant to the analysis these texts, by ranking them higher.

Fake/Real prediction. Finally, in order to provide a *Real* or *Fake* label to each piece of news to verify, their respective top n ranking texts are exploited. Specifically, occurrences of *match* and *non-match* are weighted and counted, and the majority class is used to obtain the final prediction for the piece of news. As for the weighting function, simply the inverse of the rank is used to weight each occurrence, in order to assign more weight to the highest ranking elements. As for the final prediction, the class *Fake* is assigned when most of the top n ranking elements are classified as *non-match*, and vice versa the class *Real* when most of the elements are classified as *match* for the given news.

Results on the Notre Dame Fire Dataset

The proposed approach was applied in the context of the Notre Dame Fire dataset. The collection and labelling strategy described in Section 5.2, allows for the testing of the proposed fake news detection method based on fact-checking.

In fact, as for the pieces of news to be verified, the labelled texts are available. Labelled texts include both tweets and articles, randomly selected from the dataset in order to represent both classes. The labelling can be considered as *gold*, as it has been first performed via crowdsourcing and then further manually checked. As for the ground truth, social media posts and articles produced by a list of authoritative and trustworthy sources are used.

In order to rank the ground truth with respect to each news to be verified, sequence-pairs were generated. Specifically, all possible pairs of news texts and ground truths were considered. In this case, the Information Extraction step described in Section 5.1 was not performed since all the texts pertain to the Notre Dame Fire, and thus the filtering is expected to yield rather poor results. It must be noted that, concerning articles in the ground truth, their titles and their content were separately paired to all the news texts to be verified. The final classifier for the fact-checking task was then used to classify each pair as *match* and *non-match*.

Once pairs were classified, the ranking was produced for each news text. The final prediction (i.e. if the text is real or fake) was obtained based on the number of *match* and *non-match* classes, weighted by their rank. The news was classified as *Real* if matches outweighs non-matches, and *Fake* otherwise. Table 5.6 reports the results in terms of Precision, Recall, F1-Score and Accuracy by considering the original label given by the annotators and the predicted label obtained by classifying texts with respect to matches and non-matches in the ground truth. Results are reported for varying sizes of n used to filter on the top-ranking texts.

top-n	class	Precision	Recall	F1-Score	Accuracy
5	Fake	0.31	0.51	0.39	0.51
	Real	0.7	0.51	0.59	
	Weighted avg	0.58	0.51	0.53	
10	Fake	0.33	0.53	0.41	0.53
	Real	0.72	0.52	0.60	
	Weighted avg	0.60	0.53	0.54	
50	Fake	0.33	0.52	0.40	0.53
	Real	0.72	0.54	0.61	
	Weighted avg	0.60	0.53	0.55	
100	Fake	0.34	0.51	0.41	0.55
	Real	0.72	0.56	0.63	
	Weighted avg	0.60	0.55	0.56	
1000	Fake	0.60	0.33	0.43	0.73
	Real	0.75	0.90	0.82	
	Weighted avg	0.71	0.73	0.70	
All	Fake	0.52	0.18	0.26	0.70
	Real	0.72	0.93	0.81	
	Weighted avg	0.66	0.70	0.64	

Table 5.6: Results of classification using the fact-checking based method.

Several interesting remarks can be made by observing the obtained results. It is

clear that, while the proposed method cannot obtain state-of-the-art performances on the datasest, is still able to provide good performances.

Varying the size of n drastically affects performances. Specifically, by increasing its size from 5 to 100 all the metrics improve. The best overall results in terms of Accuracy and weighted F1-Score are obtained for $n = 1000$. However, in this case there is a noticeable decrease in the reported Recall for the Fake class, that decreases as more ground truth elements are considered for providing the class. Arguably, the Recall metric is rather important in this case, as we are mostly interested in properly recognizing fake news.

It must also be noted that performance metrics for the Real class are noticeably higher for all sizes of n with respect to the Fake ones. This is probably due to two main factors. First, the majority class in the dataset is the Real one. Second, the original fact-checking model was specifically tuned to recognize matching pairs. Thus, it is possible that the model, initially trained on several different topics to identify potential matching pairs, may have a bias towards predicting a *match* when encountering documents pertaining to the same topic. For example, all the texts in the dataset mention *Notre Dame* in some capacity, either as a hashtag or as the exact string. Thus, the model may tend to overestimate their similarity. However, we can argue that, even in this case, the pre-trained model is able to achieve promising results when applied to a rather different end goal and on a completely different dataset. This means that it is able to generalize well on the task, independently from the dataset.

In addition to this, it is interesting to notice that the data provided for the fact-checking task and the data employed in the fake news detection task have one key difference. The model was trained to classify short sequence pairs of tweets and simple claims of one or two sentences. On the contrary, the texts contained in the Notre Dame Fire dataset are often longer, including entire news articles both in the ground truth and the Real and Fake texts. Therefore, it would be interesting to assess how the length of the input may affect the final performances on the different dataset. In order to do so, a simple experiment was performed. All the entire articles were excluded from the analysis, and only pairs of tweet-tweet and tweet-title were considered. The experiment was conducted by considering $n = 100$. It is interesting to notice how, in this case, performances on all metrics improve, albeit slightly. Specifically, the weighted average Precision, Recall, and F1-Score are respectively 0.64, 0.57, and 0.59. Arguably, the experiment shows that, while the model performs better as the data are more similar to the one used for training, it is nonetheless able to work also with longer sequences.

From a more general standpoint, we believe that the proposed approach is promising in terms of the adopted strategy. It showed that the problem of fake news and fact-checking are closely related, and that strategies and models implemented for the latter can provide an invaluable help also when tackling the former. In this re-

gard, it is also interesting to notice how the development of a fine-tuning strategy that is able to take into account the semantic properties of words via pre-training of language models, and to model specific aspects of the problem of automatically verifying statements, such as the similarities between claims and ground truth information, is key to successfully perform automated fact-checking, and can provide strong insights into the identification of the veracity of news. Such fine-tuning, when applied to a completely different domain and task, in conjunction with a more refined selection strategy, was nonetheless able to provide promising performances without the need for further domain or task specific training. This may be invaluable in real world contexts because of two main reasons. First, by developing strong models that are able to generalize on the problem of fact-checking, no additional labelling of data is needed to transfer such knowledge onto the fake news domain. Second, the computational cost of training models such as the Transformer is order of magnitude higher than the one needed for performing inference. While only machines provided with GPUs are able to perform training in reasonable time, more and more devices are available, also to the end users, able to successfully employ already trained machine and deep learning models in a number of tasks.

However, several drawbacks and potential ways to address them must be taken into account. First, it is clear that the obtained results are sub-optimal. It is likely that, for example, by training a machine learning model to recognize real and fake news in the dataset, better classification results could be obtained.

Second, as the original goal of the model was to recognize potential matches, its training reflected this idea. In fact, for the original training, described in Section 5.1, only one negative example was provided for each correct match. This may have had the effect to bias the classifier towards predicting matches, dampened in the fact-checking task by the fact that only the first 5 predictions ranked by probability of class *match* were taken into account. Consequently in the fake news detection task, it is possible that the model may tend to predict a match, thus degrading performances especially on the *Fake* class. In this regard, it could be interesting to fine-tune the fact-checking model on more instances, especially of non-matching pairs, in order to learn also to distinguish correct matches among many non-correct ones, in addition to learning what are the properties of correct matches. This may improve the performances of the model when dealing with identifying if statements are verified by a ground truth, rather than what elements of the ground truth verify statements.

Third, it is clear that the count-based methodology proposed to determine whether a piece of news is real or fake should be thoroughly evaluated. Specifically, more weighting schemas should be taken into account, both for the ranking and the resulting class. For example, a more strict evaluation could assign higher weights to items identified as a correct *match*, thus potentially assigning the *Real* class also to

news that have fewer supporting statements in the ground truth.

5.4 Future challenges

The problem of fake news, rumours and their related tasks is clearly an open issue. Many different aspects of the problem have been taken into account from a research perspective, such as automated fact-checking, rumour and fake news detection, verification, and tracking, and stance detection. The NLP community is thriving with works on such problems, often achieving promising results. However, the complexity of the matter at hand, and more generally the many facets assumed by misinformation and disinformation, especially on the web and social media, make it hard to provide definitive statements and clear solutions to the problem.

The work presented in this Chapter was focused on two main aspects. On the one hand, the goal was to propose several solutions to some of the major challenges in the field, namely automated fact-checking, fake news detection, and the collection and labelling of real-world datasets of fake news that can also serve as benchmark for the research community. On the other hand, the research addresses how the different aspects of the problem are related to each other, and how knowledge on one of them can be successfully transferred and exploited for others. Despite the promising results obtained, it is clear that many areas can be improved to tackle several different challenges.

Concerning aspects related to automated fact-checking, the proposed methodology has been proven to be successful, obtaining good results on the task of verified claim retrieval. In this regard, it would be interesting to extend the methodology by first and foremost incorporating more data into the training set, in order to allow for a model that is better to generalize the problem. Second, the learning phase could be improved by performing parameter and hyperparameter tuning on both the model and the selection of training candidates. Specifically for the selection of candidates, it would be interesting to consider a more unbalanced dataset of matching and non-matching pairs. Provided that the learning objective is tuned accordingly, it may yield better results for both the tasks of claim retrieval and claim verification.

As for the fake news detection problem, the approach showed promising performances when faced similarly to a claim verification task. Interestingly, it was shown that a method that does not rely on supervised learning directly from the data itself is nonetheless able to distinguish between real and fake news in some capacity. However, it is clear that there is room for improvement both in terms of approach and classification. As for the learning model, it could benefit from the same treatment previously mentioned for the automated fact-checking task. As for the classification strategy, it is obvious that more solutions should be evaluated in order to (i) improve the performances of the classifier and (ii) enable potential end

users to make more informed decision. For example, a confidence score regarding the decision could be provided, and a measure of the suspiciousness of the news in terms of the number of items in the ground truth that are in stark contrast with it, or on the contrary verify parts of it. Moreover, it could be interesting to exploit a similar methodology and learning model in a completely unsupervised scenario, in order to provide an alternative to fake news detection that is solely based on the meaning of texts.

Considering the data collection methodology, it proved to be promising since it enables the creation and labelling of a dataset of fake news that also provides a context of real news, or ground truth, for them. This may be an invaluable resource, as most available datasets do not take into account this kind of information. Concerning this aspect, it would be interesting to test state-of-the-art learning models on the dataset, in order to assess its quality as a benchmark. In addition, it would be important to study strategies for automated labelling of the rest of the dataset, for example by employing distant supervision and active learning. The methodology could be employed for the collection of real and fake news for different events and public figures, in order to provide a wider array of examples and enable learning models to generalize better. Moreover, the two-step labelling strategy may provide many interesting insights into how fake news are perceived by people and by machine learning algorithms.

Finally, the experiments performed have shown how different aspects of fake news and rumours can be considered simultaneously and how solutions to one of the many problems in the realm of fake news can be successfully adapted to other aspects of it. In this regard, pushing this line of research forward by proposing solutions that consider these many aspects simultaneously, and the interplay between them, could also be a novel and interesting point of view in the literature on fake news and rumour detection.

5.5 Summary

In this Chapter, the problem of fake news detection is faced from several different, yet interconnected, perspectives.

In Section 5.1, a system to perform the automatic retrieval of claims that verify statements is proposed. The system exploits traditional information extraction techniques and state-of-the-art language models to solve the problem. Specifically, a Sentence-BERT model is further fine tuned with two cascading steps. In the first one, the semantic similarity of matching and non matching pairs is learned in order to allow a more similar distributed representation for matching pairs. In the second one, a classifier is trained to actually provide a *match* or *non-match* label for sequence pairs. The classifier is exploited to rank verified claims for tweets, in order

to rank the claim that verifies the tweet as high as possible. The proposed method was tested in an international fact-checking shared task obtaining the second-best result (Passaro et al., 2020).

In Section 5.2 a methodology for collecting and labelling fake and real news for specific events is proposed. The collection methodology is based on a top-down strategy to collect real and fake news for an event, and news from trustworthy sources that can serve as a ground truth for the event. The labelling methodology is based on crowdsourcing, and enables a gold-level annotation of fake and real news, and the identification of fake news that are particularly difficult to be recognized by people without prior knowledge on them. Using this methodology, a dataset concerning the Notre Dame fire of April 2019 was collected and labelled considering several different fake news.

Section 5.3 provides some early insight into the adaptation of the fact-checking system proposed in Section 5.1 in order to solve the fake news detection task. The proposed system exploit the classification results provided by the fact-checking model in order to determine whether a piece of content (tweet or article) is likely to be a fake news based on its comparison with a ground truth. The system is evaluated on the Notre Dame Fire dataset (see Sec. 5.2).

Finally, Section 5.4 provides a brief discussion on the proposed methods and potential further improvement to them, as well as possible directions for the research in this area.

Chapter 6

Discussion

The experiments discussed in the present work provide many interesting insights on several aspects of computational models of language. The last few years have seen some major breakthroughs in the representation of language, and consequently the interest from many other research and industrial fields in exploiting such kind of knowledge. Therefore, it is interesting to discuss several aspects that are related to the application of such language model technologies across different problems. In this Chapter, the proposed methods and performed experiments will be briefly discussed in light of the obtained results.

One key aspect that unites all the proposed methods and several of the experiments performed in this thesis is the use of pre-trained models of language, albeit in different forms. This enabled to evaluate aspects that are common to most, if not all language models, especially from an application perspective, and several key differences among them.

Pre-trained language models have first and foremost shown very good performances when applied across a wide spectrum of tasks and domains. Simpler tasks such as obtaining distributed representation of words out-of-context have been extensively studied in the literature, and the use of pre-trained models has been proven very reliable. The experiments conducted in Chapter 3 serve as further proof of the ability of pre-trained word embedding models to encode semantically rich information for words. Specifically, learning models based on the skip-gram and C-BOW algorithms proposed in Mikolov et al. (2013a,b) have shown to enable unsupervised learning on word embeddings. In the specific case study, fastText pre-trained embeddings (Bojanowski et al., 2017) for tags of news articles were clustered in order to find macro-categories of them that enable the profiling of unseen news articles. The obtained cluster were both cohesive and meaningful, thus proving the effectiveness of the pre-trained representation.

In the case of modelling longer sequences of texts, more refined algorithms are needed. Experiments conducted in Chapter 4 provide some insights on this specific

issue. Several different methods for representing entire sentences or, more generally, longer sequences of texts, have been explored and validated.

First, it was made clear that models based on out-of-context word representations such as fastText (Bojanowski et al., 2017) were completely ineffective when applied to longer and domain-specific sequences of texts. The simplest approach to model sequences is in fact to exploit pre-trained word embeddings in order to obtain a representation for each element of the sequence, and then apply a transformation such as averaging each dimension in order to extract a representation of the whole sequence. Section 4.1 has shown that this method is rather inefficient and has severe limitations. In fact, the obtained sequence representations were shown to not be meaningful in the context of identifying similarities between text. This is probably because the pooling operation is applied to very different word vectors, that are not in the same portion of the n-dimensional space of representation. Therefore the resulting representations, also for very different texts, were rather close in the vector space, thus eliminating the possibility of identifying similarities and differences by means of comparing such vectors in terms of their relative position in the vector space.

Second, the possibility of implementing models that can learn directly from the available data such as doc2vec (Le and Mikolov, 2014) was shown to provide two key advantages over pre-trained word embeddings.

- They are specifically aimed at building representations for word sequences, that take into account the terms included in the representation. This in turn enables to model sequences such as documents more effectively, and compare them in terms of similarity of the resulting vector representation.

- As they learn directly from data, they appear to be more suitable to model domain-specific data. This advantage may be crucial when considering application of language models, especially if the end application is expected to effectively model data in a specific domain or format such as résumés. In fact, it is often the case that pre-trained models are learned on corpora containing general purpose texts, such as for example news articles, web pages, and Wikipedia entries. However, it must be noted that this crucial advantage may be severely hindered by the fact that learned models are rather specific, and thus less able to generalize on other domains or different kinds of sequence data. In addition to this, learning is expensive both in terms of time and computational resources, as already mentioned in Mikolov et al. (2013a) and shown in Section 4.1 for the problem of learning résumés embeddings. This may pose problems especially when dealing with non-static data and datasets, that evolve over time and require models to be further trained and kept up-to-date.

Third, the models based on the strong-pre training proposed for BERT (Devlin et al., 2019) have been proven more effective on most NLP downstream tasks, thus overcoming several limitations derived from representing domain specific data. Moreover, the transformer-based pre-trained model proposed in Reimers and Gurevych (2019) addresses the limitation of the sentence representation provided by BERT-like models by applying a fine tuning specifically aimed at modelling entire sequences in the vector space. The resulting sequence vectors were shown to be semantically relevant in the experiments, thus enabling a more refined comparison of them in a vector space via clustering analysis. Their pre-trained nature is in this context extremely relevant also due to the fact that, from an application perspective, the base model can be successfully exploited without the need for further training on the specific data, provided that the goal is to extract semantically relevant features from entire sequences of text.

Considering this last point, it is also important to stress how such pre-trained model can be further tuned on specific tasks such as classification. The capability of Transformer-based models to achieve transfer learning for NLP task is definitely a fundamental milestone in the development of new models and novel applications for language technologies. In the present work, transfer learning was applied to the tasks of automated fact-checking and fake news detection in Chapter 5, obtaining impressive results. The key advantage of applying transfer learning is that pre-trained model can be fine tuned to solve a wide array of downstream tasks. Language models with a strong pre-training, that can effectively model different kinds of semantic and syntactic information, can pass that knowledge also to models focused on specific downstream tasks. This approach provide several key advantages.

- Resulting models have provided state-of-the-art performances in a vast number of NLP tasks.
- They do not require vast amounts of data to be effectively trained on downstream tasks. This is because the model, rather than learning word properties from scratch in the context of the specific problem, is only tasked to learn the task based on the general language knowledge acquired during pre-training. This also means that the same pre-trained model can be fine tuned both on different tasks and in order to solve many tasks at the same time. From an application perspective, this may prove to be extremely important, as task-specific training is less expensive in terms of required learning data, and as the same model can be used for several different tasks. In the present research, a model trained for a fact-checking task was successfully applied to the similar, yet different, fake news detection task.

As for the problem of fake news detection, several important remarks can be made concerning the approach both in terms of performances and with respect to

the current state of the research on this topic.

Considering the approach itself, Section 5.1 showed that it has proven to be very effective on the verified claim retrieval task it was trained for. First, the final model with two-step cascade fine-tuning strategy was shown to obtain MAP@5 of around 0.90 on the task. This can lead to two considerations. On the one hand, it is interesting to notice how the first fine tuning step aimed at obtaining representations that are closer in the vector space for matching pairs had a rather drastic effect on the final prediction. On the other hand, we can argue that modelling a ranking problem in terms of classification was nonetheless effective. In addition to this, it is also important to stress on the effect of fine-tuning a pre-trained Sentence-BERT model rather than a more general BERT one. This serves as further proof that sequence embeddings obtained by BERT models are not modelling the semantics of sequences in any meaningful way (Devlin et al., 2019), but can be easily adapted to the task by fine tuning the model on sequence-pair similarity tasks (Reimers and Gurevych, 2019). Second, the proposed IE module was able to identify claims that are more likely to verify a given tweet. By applying this simple system, the performances of the whole system improved. This is probably due to the fact that it was effective in reducing the negative impact of the training strategy for the model. In fact, the final training of the classifier was performed on a balanced dataset of correct and incorrect pairs, while during inference the incorrect pairs are orders of magnitude more than the correct ones. By simply filtering out pairs that do not share any meaningful entity, errors due to incorrect classification on them were avoided.

In addition to this, by considering the experiment described in Section 5.3, two main observations can be made. On the one hand, it was shown how the model trained for fact-checking obtained encouraging performances also on the fake news detection task with no further training. This means that the original model was already able to generalize to the problem of matching sentences with similar or identical meanings. On the other hand, the experiment provides some insight into how approaches aimed at fact-checking statements can be effectively applied to fake news detection problems. In fact, albeit several improvements can be made to the proposed approach, it is clear that it is nonetheless promising. In addition, as it implements, both conceptually and during training, a strategy that relies on the actual meaning of words and sentences rather than using more surface properties that may have limited descriptive capabilities, it is expected to generalize better on other data and in real-world scenarios.

Concerning the current state of the research on the fake news topic, we can argue that the present work paves the way to providing solution with the potential to prove themselves as viable alternatives to current approaches. First, as previously mentioned, the fact-checking based strategy has proven reliable in order to identify real and fake news.

Second, the methodology for data collection and labelling described in Section 5.2 provides several interesting differences from current methods, highlighted in Section 2.2. First, it offers a simple yet effective approach to the collection itself, based on accessible social media data. Second, it provides a two-layered labelling that may help in better describing the fake news problem, especially concerning more hard-to-detect ones. Third, and most important, it incorporates into the collection strategy also the retrieval of the context in which fake news propagate. We can argue that this inclusion may prove to be fundamental. On the one hand, it operationally provides for a ground truth against which to compare possible real and fake news. On the other hand, it helps in representing more real-word scenarios where fake news are actually hidden among many other kinds of information. This can be helpful for evaluating fake news detection systems, both from the perspective or benchmark dataset for scientific research, and for real-world applications.

Chapter 7

Conclusions

The goal of the present work was twofold. On the one hand, it aimed at evaluating several different approaches to distributional semantics in the context of applications for NLP, in order to better assess their strengths, weaknesses, and key differences. On the other hand, such knowledge was exploited in order to provide novel insights and strategies to face the problem of fake news and rumour detection, that can be considered one of the main current issues both at the social level and from a research perspective.

As for the evaluation of distributional semantics models, they hold a key strategic value in many modern applications, especially considering the constantly growing need for methods and models that are able to effectively and efficiently process unstructured data. This has in fact become one of the main driving forces for many application-oriented researches in the NLP field, also due to the fact that such unstructured information holds a key value for businesses and companies in the wake of Industry 4.0. The ability to extract usable knowledge directly from data with minor supervision, and automate and speed-up processes that can be consuming and expensive both in terms of time and resources, has become an invaluable asset for all the sectors influenced by the ideas and concepts of Industry 4.0.

In the present work, distributional models of meaning, in conjunction with a mixture of NLP and text mining techniques, have been applied to several different case studies, in order to evaluate their key characteristics and their effectiveness when used for tasks of profiling in different areas of application, with particular attention to pre-trained models. First, in Chapter 3 a pre-trained fastText model in conjunction with an SVM classifier were used to profile city areas based on reports contained in online newspapers. Specifically, the tags describing the articles were used to provide several macro-categories based on the application of hierarchical clustering to their word embeddings. Such macro-categories were then used to label articles and train a multi-class classifier in order to determine the macro-category of new articles. The resulting models were further incorporated in a framework for profiling

specific neighborhoods in terms of the macro categories, based on geo-localization and the news articles describing them. Results show that the proposed approach is promising, both in terms of understanding the similarities between words, and thus representing macro-categories, and in terms of obtaining reliable predictions from the classification. Subsequently, in Chapter 4 the analysis shifted towards distributional models aimed at providing a distributed representation for entire sequences of texts. Arguably such models, despite facing the more challenging problem of providing a semantic representation for larger units of text, may also provide better solutions for more complex and challenging problems, such as the profiling of entire unstructured texts in an unsupervised way. In order to evaluate the best approach, a case study on the profiling of résumés was proposed. In the case study, several techniques were evaluated, such as pre-trained word embeddings models, the paragraph vector algorithm, and Transformer-based language models. The analysis was conducted in conjunction with more traditional techniques of information extraction, to obtain keywords from texts, and summarization techniques aimed at shortening the length of résumés in order to obtain more cohesive and focused texts, that are in turn easier to model distributionally. Résumés, modelled as distributed vectors of words, were applied hierarchical clustering in order to identify the different professional profiles. Results were evaluated both qualitatively and quantitatively, and showed promise. The findings of Chapter 3 and 4 made it clear that pre-trained transformer-based models were able to outperform other distributional techniques, and provided several advantages also from the point of view of the end application, as discussed in Chapter 6.

As for the problem of fake news detection, the knowledge obtained from the case studies on distributional semantic models was put to the test by proposing a strategy that focuses on fact-checking to identify trustworthy and not trustworthy information. Specifically, as described in Chapter 5, first a system for fact-checking based on state-of-the-art Transformer language models was proposed. The system goal is to identify, for a given statement, a claim provided in a ground truth that verifies it. To this end, a Transformer model was fine tuned with cascading training steps, in conjunction with an Information Extraction system to improve the results. The final model predicts, given a pair of texts, if the second one (a verified claim) actually verifies the information contained in the first one (a statement or tweet). The model obtained second-best results in a fact-checking competition. In order to evaluate the relationship between fact-checking and fake news detection, and to further evaluate the model, a dataset of real and fake news concerning the Notre Dame Cathedral fire of 2019 was collected. The dataset includes both a ground truth provided by reliable news sources, and a set of news in the form of tweets and news articles labelled via crowdsourcing for their truthfulness. The proposed methodology allow to collect and annotate reliable data concerning real and fake news for specific events or public

figures based on Twitter. In addition, it allows for labelling fake news with respect to how easy they are to identify without prior knowledge about them. The methodology is believed to be a viable strategy to collect and label benchmark datasets for fake news research. The fact-checking model was applied to the Notre Dame fire dataset in order to distinguish real and fake news based on their content with respect to the available ground truth. Although with wide margins for improvement, the obtained results allowed to state that aspects of fact-checking, and specifically of claim retrieval, can improve also fake news detection, and that the direction of the research is promising. It was possible to demonstrate how a system based on the smart usage of distributional models of meaning, that is able to understand and model the differences in information contained in trustworthy and non-trustworthy pieces of news, provided a viable alternative to facing the problem of fake news detection with respect to a more traditional classification-based approach.

Language technologies are becoming more and more pervasive in many fields of research and in industrial settings. The recent years have seen the introduction of models able to provide transfer learning capabilities, that can arguably be considered revolutionary in the field. Such models have proven their worth in a number of settings. This is demonstrated also by the experiments and case studies provided in the present work, specifically considering those regarding fake news detection. Arguably, as research on such models is fueled by improving the performances on the one hand, and reducing the size of models to make them usable with less compute on the other one, it is clear that their understanding and intelligent use can prove to be an invaluable asset both from the research and from the industrial standpoint, to face increasingly complex problems that are hard to model and even harder to solve also for humans, such as the fake news detection one. Being able to automatically and unsupervisedly provide information regarding the potential veracity or falsity of information may prove to be an fundamental stepping stone in fighting disinformation and many other issues in today's society.

Definitely, the present work tried to demonstrate that modelling the meaning of textual information allows to deal with several real-world problems successfully.

Appendix A

Publications

Journal papers

1. A. Bechini, **A. Bondielli**, P. Ducange, F. Marcelloni and A. Renda, "Addressing Event-Driven Concept Drift in Twitter Stream: A Stance Detection Application", *IEEE Access*, vol. 9, pp. 77758-77770, 2021.
   Candidate's contributions: design and development of the semantic learning scheme; evaluation of the results.

2. **Alessandro Bondielli**, Francesco Marcelloni, "A survey on fake news and rumour detection techniques", *Information Sciences, 497(38)*, pages:38–55, 2019.
   Candidate's contributions: Review of the literature on the different topics concerned by the survey; evaluation; paper writing.

3. Lucia C. Passaro, **Alessandro Bondielli**, Alessandro Lenci, "Learning affect with Distributional Semantic Models", *Italian Journal of Computational Linguistics, 3(2)*, pages:23–36, 2017.
   Candidate's contributions: design, implementation and practical evaluation of the proposed experiments.

4. Gianluca E. Lebani, **Alessandro Bondielli**, and Alessandro Lenci "You are what you do. An empirical characterization of the semantic content of the thematic roles for a group of Italian verbs", *Journal of Cognitive Science, 16(4)*, pages:401–430, 2015.
   Candidate's contributions: design and implementation of the platform for the collection of norms; data collection.

Peer reviewed conference papers

1. **Alessandro Bondielli**, Pietro Ducange, Francesco Marcelloni, "Exploiting Categorization of Online News for Profiling City Areas", *Proceedings of the 2020*

IEEE Conference on Evolving and Adaptive Intelligent Systems (*EAIS*),pages:1–8, 2020.
Candidate's contributions: supervision on the experiment; paper writing.

2. **Alessandro Bondielli**, Francesco Marcelloni, "A Data-Driven Approach to Automatic Extraction of Professional Figure Profiles from Résumés", *Intelligent Data Engineering and Automated Learning* (*IDEAL 2019*), pages:155–165, 2019.
Candidate's contributions: theoretical analysis of the problem; design of the algorithms and the performed experiments; evaluation of the results; paper writing.

3. **Alessandro Bondielli**, Lucia C. Passaro, Alessandro Lenci, "CoreNLP-it: A UD Pipeline for Italian based on Stanford CoreNLP", *Proceedings of the Fifth Italian Conference on Computational Linguistics* (*CLiC-it 2018*),pages:57–61, 2018.
Candidate's contributions: development of the algorithms and integration in the CoreNLP framework; performance evaluation; paper writing.

4. **Alessandro Bondielli**, Lucia C. Passaro, Alessandro Lenci, "Emo2Val: Inferring Valence Scores from fine-grained Emotion Values", *Proceedings of the Fourth Italian Conference on Computational Linguistics* (*CLiC-it 2017*), pages:48–52, 2017.
Candidate's contributions: design, implementation and practical evaluation of the experiments; paper writing.

5. Lucia C. Passaro, **Alessandro Bondielli**, Alessandro Lenci "FB-NEWS15: A Topic-Annotated Facebook Corpus for Emotion Detection and Sentiment Analysis", *Proceedings of the Third Italian Conference on Computational Linguistics* (*CLiC-it 2016*), pages:228–232, 2016.
Candidate's contributions: design and implementation of the algorithm for data collection; data collection and annotation.

Workshop papers

1. **Alessandro Bondielli**, Gianluca E. Lebani, Lucia C. Passaro, Alessandro Lenci, "CAPISCO @CONcreTEXT 2020: (Un)supervised Systems to Contextualize Concreteness with Norming Data", *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop* (*EVALITA 2020*), 2020.
Candidate's contributions: design of the algorithms and part of their implementation; paper writing.

2. Lucia C. Passaro, **Alessandro Bondielli**, Alessandro Lenci, Francesco Marcelloni, "UNIPI-NLE at CheckThat! 2020: Approaching Fact Checking from a Sentence Similarity Perspective Through the Lens of Transformers", *Working*

Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, 2020.
Candidate's contributions: theoretical analysis; design of the algorithm.

3. Lucia C. Passaro, **Alessandro Bondielli**, Alessandro Lenci. "Exploiting Emotive Features for the Sentiment Polarity Classification of tweets", *EVALITA. Evaluation of NLP and Speech Tools for Italian: Proceedings of the Final Workshop*, pages:205–210, 2016.
Candidate's contributions: feature design; carried out the experiments.

Papers under review

1. Lucia C. Passaro, **Alessandro Bondielli**, Pietro Dell'Oglio, Alessandro Lenci, Francesco Marcelloni, "In-context annotation of Topic-Oriented Datasets of Fake News: A Case study on the Notre-Dame Fire Event".
Candidate's contributions: co-design of the collection and annotation methodology; classification experiments and evaluation.

2. Alessio Bechini, **Alessandro Bondielli**, José Luis Corcuera Bárcena, Pietro Ducange, Francesco Marcelloni, Alessandro Renda, "Mining the Stream of News for City Areas Profiling: a Case Study for the City of Rome".
Candidate's contributions: design and supervision of the experiment; paper writing.

3. **Alessandro Bondielli**, Francesco Marcelloni, "On the use of Summarization and Transformer Architectures for Profiling Résumés".
Candidate's contributions: design of the method and the experiments; implementation of the algorithms; experimentation and evaluation of the method; paper writing.

Bibliography

Afroz, S., Brennan, M., and Greenstadt, R. (2012). Detecting hoaxes, frauds, and deception in writing style online. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 461–475, Washington, DC, USA. IEEE Computer Society.

Agirre, E. and Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. EACL '09, page 33–41, USA. Association for Computational Linguistics.

Ajao, O., Bhowmik, D., and Zargari, S. (2018). Fake news identification on twitter with hybrid cnn and rnn models. In *Proceedings of the 9th International Conference on Social Media and Society*, SMSociety '18, pages 226–230, New York, NY, USA. ACM.

Aker, A., Derczynski, L., and Bontcheva, K. (2017). Simple open stance classification for rumour analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 31–39.

Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research.

Allport, G. W. and Postman, L. (1946). An analysis of rumor. *Public Opinion Quarterly*, 10(4):501–517.

Allport, G. W. and Postman, L. (1947). The psychology of rumor. *The ANNALS of the American Academy of Political and Social Science*, 257(1):240–241.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486.

Anastasi, G., Antonelli, M., Bechini, A., Brienza, S., D'Andrea, E., De Guglielmo, D., Ducange, P., Lazzerini, B., Marcelloni, F., and Segatori, A. (2013). Urban and social sensing for sustainable mobility in smart cities. In *Proceedings of the 2013 Sustainable Internet and ICT for Sustainability Conference(SustainIT)*, pages 1–4, Palermo, Italy.

Ando, R. K. (2000). Latent semantic space: Iterative scaling improves precision of inter-document similarity measurement. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 216–223, New York, NY, USA. Association for Computing Machinery.

Aouicha, M. B. and Taieb, M. A. H. (2015). G2ws: Gloss-based wordnet and wiktionary semantic similarity measure. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–7.

Arampatzis, A., Kanoulas, E., Tsikrika, T., Vrochidis, S., Joho, H., Lioma, C., Eickhoff, C., Névéol, A., Cappellato, L., and Ferro, N., editors (2020). *Experimental IR Meets Multilinguality, Multimodality, and Interaction Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020)*, LNCS (12260). Springer.

Bagga, A. and Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, San Diego, CA, USA.

Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, page 805–810, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Barrios, F., López, F., Argerich, L., and Wachenchauzer, R. (2016). Variations of the Similarity Function of TextRank for Automated Summarization. *arXiv e-prints*.

Barrón-Cedeño, A., Elsayed, T., Nakov, P., Da San Martino, G., Hasanain, M., Suwaileh, R., and Haouari, F. (2020). Checkthat! at clef 2020: Enabling the automatic identification and verification of claims in social media. In Jose, J. M.,

Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M. J., and Martins, F., editors, *Advances in Information Retrieval*, pages 499–507, Cham. Springer International Publishing.

Barrón-Cedeño, A., Elsayed, T., Nakov, P., Da San Martino, G., Hasanain, M., Suwaileh, R., Haouari, F., Babulkov, N., Hamdan, B., Nikolov, A., Shaar, S., and Sheikh Ali, Z. (2020). Overview of CheckThat! 2020: Automatic identification and verification of claims in social media. In Arampatzis et al. (2020).

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3(null):1137–1155.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(null):993–1022.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bondielli, A. and Marcelloni, F. (2019). A survey on fake news and rumour detection techniques. *Information Sciences*, 497:38 – 55.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Brand, M. (2006). Brand, m.: Fast low-rank modifications of the thin singular value decomposition. linear algebra appl. 415(1), 20-30. *Linear Algebra and its Applications*, 415:20–30.

Briscoe, E. J., Appling, D. S., and Hayes, H. (2014). Cues to deception in social media communications. In *Proceedings of the 2014 47th Hawaii International Conference on System Sciences (HICSS)*, pages 1435–1443. IEEE.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Bullinaria, J. A. and Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.

Buntine, W. and Jakulin, A. (2006). Discrete component analysis. In Saunders, C., Grobelnik, M., Gunn, S., and Shawe-Taylor, J., editors, *Subspace, Latent Structure and Feature Selection*, pages 1–33, Berlin, Heidelberg. Springer Berlin Heidelberg.

Cai, G., Wu, H., and Lv, R. (2014). Rumors detection in chinese via crowd responses. In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 912–917, Beijing, China. IEEE.

Camacho-Collados, J., Pilehvar, M. T., and Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36 – 64.

Canini, K. R., Suh, B., and Pirolli, P. L. (2011). Finding credible information sources in social networks based on content and social structure. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 1–8. IEEE.

Cappellato, L., Eickhoff, C., Ferro, N., and Névéol, A., editors (2020). *Working Notes of CLEF 2020—Conference and Labs of the Evaluation Forum*.

Castillo, C., Mendoza, M., and Poblete, B. (2011). Information credibility on twitter. In *Proceedings of the 20th international conference on World Wide Web*, pages 675–684, Hyderabad, India. ACM.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Chang, C., Zhang, Y., Szabo, C., and Sheng, Q. Z. (2016). Extreme user and political rumor detection on twitter. In *Proceedings of the 12th International Conference on Advanced Data Mining and Applications(ADMA)*, pages 751–763. Springer.

Chen, Y.-C., Liu, Z.-Y., and Kao, H.-Y. (2017). Ikm at semeval-2017 task 8: Convolutional neural networks for stance detection and rumor verification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 465–469.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chua, A. Y. K. and Banerjee, S. (2016). Linguistic predictors of rumor veracity on the internet. In *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)*, pages 387–391.

Church, K. W. and Hanks, P. (1989). Word association norms, mutual information, and lexicography. In *27th Annual Meeting of the Association for Computational Linguistics*, pages 76–83, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., and Flammini, A. (2015). Computational fact checking from knowledge networks. *PloS one*, 10(6).

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. Curran Associates, Inc.

Conroy, N. J., Rubin, V. L., and Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, 52(1):1–4.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

D'Andrea, E., Ducange, P., Bechini, A., Renda, A., and Marcelloni, F. (2019). Monitoring the public opinion about the vaccination topic from tweets analysis. *Expert Systems with Applications*, 116:209–226.

D'Andrea, E., Ducange, P., Lazzerini, B., and Marcelloni, F. (2015). Real-time detection of traffic from twitter stream analysis. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2269–2283.

D'Andrea, E., Ducange, P., Loffreno, D., Marcelloni, F., and Zaccone, T. (2018). Smart profiling of city areas based on web data. In *2018 IEEE International Conference on Smart Computing*, pages 226–233. IEEE.

Das, A., Yenala, H., Chinnakotla, M., and Shrivastava, M. (2016). Together we stand: Siamese networks for similar question retrieval. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 378–387, Berlin, Germany. Association for Computational Linguistics.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407.

Derczynski, L. and Bontcheva, K. (2014). Pheme: Veracity in digital social networks. In *Proceedings of the 10th Joint ACL – ISO Workshop on Interoperable Semantic Annotation (ISA)*, pages 19–22, Reykjavik, Iceland.

Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Wong Sak Hoi, G., and Zubiaga, A. (2017). SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

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

Di Fonzo, N. and Bordia, P. (2007). Rumor, gossip and urban legends. *Diogenes*, 54(1):19–35.

Diakopoulos, N., De Choudhury, M., and Naaman, M. (2012). Finding and assessing social media information sources in the context of journalism. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2451–2460. ACM.

Dong, L., Srimani, P. K., and Wang, J. Z. (2010). West: Weighted-edge based similarity measurement tools for word semantics. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, page 216–223, USA. IEEE Computer Society.

Dumais, S. T. (1992). Enhancing performance in latent semantic indexing (lsi) retrieval.

Eckhardt, A., Laumer, S., Maier, C., and Weitzel, T. (2014). The transformation of people, processes, and it in e-recruiting : Insights from an eight-year case study of a german media corporation. *Employee Relations*, 36.

Elsayed, T., Nakov, P., Barrón-Cedeño, A., Hasanain, M., Suwaileh, R., Da San Martino, G., and Atanasova, P. (2019). Checkthat! at clef 2019: Automatic identification and verification of claims. In Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., and Hiemstra, D., editors, *Advances in Information Retrieval*, pages 309–315, Cham. Springer International Publishing.

Erk, K. and Padó, S. (2008). A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii. Association for Computational Linguistics.

Ettinger, A. (2020). What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. The MIT Press.

Fellbaum, C. (2005). Wordnet and wordnets. In Barber, A., editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier.

Feng, V. W. and Hirst, G. (2013). Detecting deceptive opinions with profile compatibility. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 338–346.

Ferreira, W. and Vlachos, A. (2016). Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1163–1168.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. 1952-59:1–32.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378—382.

Giasemidis, G., Singleton, C., Agrafiotis, I., Nurse, J. R., Pilgrim, A., Willis, C., and Greetham, D. V. (2016). Determining the veracity of rumours on twitter. In *International Conference on Social Informatics*, pages 185–205. Springer.

Giatsoglou, M., Chatzakou, D., Gkatziaki, V., Vakali, A., and Anthopoulos, L. (2016). Citypulse:a platform prototype for smart city social data mining. *Journal of the Knowledge Economy*, 7:344–372.

Goikoetxea, J., Soroa, A., and Agirre, E. (2015). Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1434–1439, Denver, Colorado. Association for Computational Linguistics.

Gorrell, G. (2006). Generalized hebbian algorithm for incremental singular value decomposition in natural language processing. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Gorrell, G., Kochkina, E., Liakata, M., Aker, A., Zubiaga, A., Bontcheva, K., and Derczynski, L. (2019). SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Guo, F.-M., Liu, S., Mungall, F. S., Lin, X., and Wang, Y. (2019). Reweighted proximal pruning for large-scale language representation. *ArXiv*, abs/1909.12486.

Gupta, A., Kumaraguru, P., Castillo, C., and Meier, P. (2014). Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*, pages 228–243. Springer.

Hamidian, S. and Diab, M. (2015). Rumor detection and classification for twitter data. In *Proceedings of the 5th International Conference on Social Media Technologies, Communication, and Informatics, SOTICS, IARIA*, pages 71–77.

Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., and Gurevych, I. (2018). A retrospective analysis of the fake news challenge stance-detection task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1859–1874, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hardalov, M., Koychev, I., and Nakov, P. (2016). In search of credible news. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 172–180. Springer.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Harshman, R. (1970). Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16.

Heggo, I. A. and Abdelbaki, N. (2018). *Hybrid Information Filtering Engine for Personalized Job Recommender System*, pages 553–563. Springer International Publishing, Cham.

Hermida, A. (2010). Twittering the news: The emergence of ambient journalism. *Journalism practice*, 4(3):297–308.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Hill, F., Cho, K., and Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 50–57, New York, NY, USA. Association for Computing Machinery.

Horne, B. D. and Adali, S. (2017). This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News. *arXiv e-prints*, page arXiv:1703.09398.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

Iglesias, J. A., Tiemblo, A., Ledezma, A., and Sanchis, A. (2016). Web news mining in an evolving framework. *Information Fusion*, 28:90–98.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. In *Association for Computational Linguistics*.

Jin, Z., Cao, J., Zhang, Y., and Luo, J. (2016). News verification by exploiting conflicting social viewpoints in microblogs. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 2972–2978.

Jin, Z., Cao, J., Zhang, Y., Zhou, J., and Tian, Q. (2017). Novel visual and statistical image features for microblogs news verification. *IEEE transactions on multimedia*, 19(3):598–608.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling.

K, K., Wang, Z., Mayhew, S., and Roth, D. (2019). Cross-lingual ability of multilingual bert: An empirical study. *arXiv preprint arXiv:1912.07840*.

Kang, C. and Goldman, A. (2016). In washington pizzeria attack, fake news brought real guns. *The New York Times*, 5.

Kim, Y., Denton, C., Hoang, L., and Rush, A. M. (2017). Structured attention networks. *ArXiv*, abs/1702.00887.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015). Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3294–3302, Cambridge, MA, USA. MIT Press.

Kochkina, E., Liakata, M., and Zubiaga, A. (2018). All-in-one: Multi-task learning for rumour verification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3402–3413.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. (2019). Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. (2015). From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org.

Kwon, S., Cha, M., Jung, K., Chen, W., and Wang, Y. (2013). Prominent features of rumor propagation in online social media. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining (ICDM)*, pages 1103–1108, Dallas, Texas, USA. IEEE.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Landauer, T. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Lastra-Díaz, J. J., Goikoetxea, J., Hadj Taieb, M. A., García-Serrano, A., Ben Aouicha, M., and Agirre, E. (2019). A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art. *Engineering Applications of Artificial Intelligence*, 85:645 – 665.

Laumer, S., Maier, C., and Eckhardt, A. (2014). The impact of business process management and applicant tracking systems on recruiting process performance: An empirical study. *Journal of Business Economics*, 85.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org.

Lee, D. and Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–91.

Lenci, A. (2018). Distributional models of word meaning. *Annual Review of Linguistics*, 4(1):151–171.

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

Likavec, S., Lombardi, I., and Cena, F. (2019). Sigmoid similarity - a new feature-based similarity measure. *Information Sciences*, 481:203 – 218.

Lin, A. Y., Ford, J., Adar, E., and Hecht, B. (2018). Vizbywiki: Mining data visualizations from the web to enrich news articles. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 873–882. Int. World Wide Web Conf.s Steering Committee.

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98/COLING '98, page 768–774, USA. Association for Computational Linguistics.

Lin, D. (1998b). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lin, Y., Tan, Y. C., and Frank, R. (2019). Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Liu, P. J., Saleh, M. A., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *Proceedings of ICLR 2018*.

Liu, X., Nourbakhsh, A., Li, Q., Fang, R., and Shah, S. (2015). Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1867–1870. ACM.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Logeswaran, L. and Lee, H. (2018). An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *In Proceedings of the 2019 International Conference on Learning Representations*.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28:203–208.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., and Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks. In *IJCAI'16 Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3818–3824, New York, NY, USA.

Ma, J., Gao, W., Wei, Z., Lu, Y., and Wong, K.-F. (2015). Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1751–1754, Melbourne, VIC, Australia. ACM.

Magdy, A. and Wanas, N. (2010). Web-based statistical fact checking of textual documents. In *Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents*, SMUC '10, pages 103–110. ACM.

Mickus, T., Paperno, D., Constant, M., and van Deemter, K. (2019). What do you mean, bert? assessing bert as a distributional semantics model. *ArXiv*, abs/1911.05758.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Miller, D. (2019). Leveraging BERT for Extractive Text Summarization on Lectures. *arXiv e-prints*, page arXiv:1906.04165.

Miller, G. A. (1995). Wordnet: A lexical database for english. 38(11):39–41.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language & Cognitive Processes*, 6(1):1–28.

Mitra, T. and Gilbert, E. (2015). Credbank: A large-scale social media corpus with associated credibility annotations. In *Proceedings of the 9th International AAAI Conference on Web and Social Media*, pages 258–267.

Mukherjee, S. and Weikum, G. (2015). Leveraging joint interactions for credibility analysis in news communities. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 353–362.

Newman, N., Dutton, W. H., and Blank, G. (2012). Social media in the changing ecology of news: The fourth and fifth estates in britain. *International Journal of Internet Science*, 7(1):6–22.

Nicosia, M. and Moschitti, A. (2017). Learning contextual embeddings for structural semantic similarity using categorical information. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 260–270, Vancouver, Canada. Association for Computational Linguistics.

Nie, Y., Chen, H., and Bansal, M. (2019). Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Niwa, Y. and Nitta, Y. (1994). Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, page 304–309, USA. Association for Computational Linguistics.

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Passaro, L. C., Bondielli, A., Lenci, A., and Marcelloni, F. (2020). Unipi-nle at check-that! 2020: Approaching fact checking from a sentence similarity perspective through the lens of transformers. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum*.

Patwardhan, S. (2006). Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *In: Proceedings of the EACL*, pages 1–8.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Pérez-Rosas, V. and Mihalcea, R. (2015). Experiments in open domain deception detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1120–1125. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv e-prints*, page arXiv:1802.05365.

Pianta, E., Bentivogli, L., and Girardi, C. (2002). Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*.

Po, L. and Rollo, F. (2018). Building an urban theft map by analyzing newspaper crime reports. In *2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization*, pages 13–18. IEEE.

Popat, K., Mukherjee, S., Strötgen, J., and Weikum, G. (2017a). Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1003–1012.

Popat, K., Mukherjee, S., Strötgen, J., and Weikum, G. (2017b). Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1003–1012.

Potthast, M., Köpsel, S., Stein, B., and Hagen, M. (2016). Clickbait detection. In *European Conference on Information Retrieval*, pages 810–817. Springer.

Qazi, M., Khan, M. U. S., and Ali, M. (2020). Detection of fake news using transformer model. In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–6.

Qazvinian, V., Rosengren, E., Radev, D. R., and Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1589–1599, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Qin, Y., Wurzer, D., Lavrenko, V., and Tang, C. (2016). Spotting rumors via novelty detection. *CoRR*, abs/1611.06322.

Quintero, R., Torres-Ruiz, M., Menchaca-Mendez, R., Moreno-Armendariz, M. A., Guzman, G., and Moreno-Ibarra, M. (2019). Dis-c: Conceptual distance in ontologies, a graph-based approach. *Knowl. Inf. Syst.*, 59(1):33–65.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30.

Radford, A. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ramisa, A., Yan, F., Moreno-Noguer, F., and Mikolajczyk, K. (2018). Breakingnews: Article annotation by image and text processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1072–1085.

Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322.

Řehůřek, R. and Sojka, P. (2010). 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. ELRA.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, page 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in bertology: What we know about how bert works. *ArXiv*, abs/2002.12327.

Rubin, V., Conroy, N., Chen, Y., and Cornwell, S. (2016). Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the NAACL-CADD2016 Second Workshop on Computational Approaches to Deception Detection*, pages 7–17, San Diego, California, USA.

Rubin, V. L., Chen, Y., and Conroy, N. J. (2015). Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Ruchansky, N., Seo, S., and Liu, Y. (2017). Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806. ACM.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2013). Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):919–931.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commununication of the ACM*, 18(11):613–620.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108:arXiv:1910.01108.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal. Association for Computational Linguistics.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., editors, *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg. Springer Berlin Heidelberg.

Shaar, S., Babulkov, N., Da San Martino, G., and Nakov, P. (2020a). That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

Shaar, S., Nikolov, A., Babulkov, N., Alam, F., Barrón-Cedeño, A., Elsayed, T., Hasanain, M., Suwaileh, R., Haouari, F., Da San Martino, G., and Nakov, P. (2020b). Overview of CheckThat! 2020 English: Automatic identification and verification of claims in social media. In Cappellato et al. (2020).

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Shehu, V. and Besimi, A. (2018). *Improving Employee Recruitment Through Data Mining*, pages 194–202. Springer International Publishing.

Shi, B. and Weninger, T. (2016). Fact checking in heterogeneous information networks. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 101–102. International World Wide Web Conferences Steering Committee.

Shiralkar, P., Flammini, A., Menczer, F., and Ciampaglia, G. L. (2017). Finding streams in knowledge graphs to support fact checking. In *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM)*, pages 859–864. IEEE.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *CoRR*, abs/1809.01286.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2020). Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3):171–188.

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

Silva, B. N., Khan, M., and Han, K. (2018). Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable Cities and Society*, 38:697–713.

Slovikovskaya, V. and Attardi, G. (2020). Transfer learning from transformers to fake news challenge stance detection (FNC-1) task. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1211–1218, Marseille, France. European Language Resources Association.

Song, C., Tu, C., Yang, C., Liu, Z., and Sun, M. (2018). CED: Credible Early Detection of Social Media Rumors. *arXiv e-prints*, page arXiv:1811.04175.

Sparck Jones, K. (1988). *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, page 132–142. Taylor Graham Publishing, GBR.

Stanchev, L. (2014). Creating a similarity graph from wordnet. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, WIMS '14, New York, NY, USA. Association for Computing Machinery.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., and de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. *CoRR*, abs/1704.07506.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S. R., Das, D., and Pavlick, E. (2019). What do you learn from context? probing for sentence structure in contextualized word representations. In *Proceedings of the International Conference on Learning Representations (ICLR 2019)*.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018). Fever: a large-scale dataset for fact extraction and verification. 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 809–819. Association for Computational Linguistics.

Tolmie, P., Procter, R., Randall, D. W., Rouncefield, M., Burger, C., Wong Sak Hoi, G., Zubiaga, A., and Liakata, M. (2017). Supporting the use of user generated content in journalistic practice. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3632–3644. ACM.

Trasarti, R., Pinelli, F., Nanni, M., and Giannotti, F. (2011). Mining mobility user profiles for car pooling. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1190–1198, San Diego, CA.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Turney, P. D. (2001). Mining the web for synonyms: Pmi-ir versus lsa on toefl. In De Raedt, L. and Flach, P., editors, *Machine Learning: ECML 2001*, pages 491–502, Berlin, Heidelberg. Springer Berlin Heidelberg.

Turney, P. D. (2008). A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, page 905–912, USA. Association for Computational Linguistics.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4):327–352.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, u., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010. Curran Associates Inc.

Vieweg, S. (2010). Microblogged contributions to the emergency arena: Discovery, interpretation and implications. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, pages 241–250. ACM.

Vlachos, A. and Riedel, S. (2014). Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22. Association for Computational Linguistics.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Volkova, S., Shaffer, K., Jang, J. Y., and Hodas, N. (2017). Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 647–653.

Vosoughi, S. (2015). *Automatic detection and verification of rumors on Twitter*. PhD thesis, Massachusetts Institute of Technology.

Vosoughi, S., Mohsenvand, M., and Roy, D. (2017). Rumor gauge: predicting the veracity of rumors on twitter. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(4):50.

Wang, S. and Terano, T. (2015). Detecting rumor patterns in streaming social media. In *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data)*, pages 2709–2715.

Wang, S., Zhou, W., and Jiang, C. (2019). A survey of word embeddings based on deep learning. *Computing*, 102:717–740.

Wang, W. Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426. Association for Computational Linguistics.

Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016). Towards universal paraphrastic sentence embeddings. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wu, K., Yang, S., and Zhu, K. Q. (2015). False rumors detection on sina weibo by propagation structures. In *Proceedings of the 2015 IEEE 31st International Conference on Data Engineering (ICDE)*, pages 651–662. IEEE.

Wu, L., Yen, I. E.-H., Xu, K., Xu, F., Balakrishnan, A., Chen, P.-Y., Ravikumar, P., and Witbrock, M. J. (2018). Word mover's embedding: From Word2Vec to document embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4524–4534, Brussels, Belgium. Association for Computational Linguistics.

Wu, Y., Agarwal, P. K., Li, C., Yang, J., and Yu, C. (2014). Toward computational fact-checking. *The Proceedings of the VLDB Endowment (PVLDB)*, 7(7):589–600.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L.,

Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yang, C., Su, G., and Chen, J. (2017). Using big data to enhance crisis response and disaster resilience for a smart city. In *Proceedings of the IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pages 504–507, Beijing, China.

Yang, F., Liu, Y., Yu, X., and Yang, M. (2012). Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 13. ACM.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.

Yu, F., Liu, Q., Wu, S., Wang, L., and Tan, T. (2017). A convolutional approach for misinformation identification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 3901–3907. AAAI Press.

Zadeh, A. H. and Moshovos, A. (2020). Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference. *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 811–824.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Zhang, H., Fan, Z., Zheng, J.-h., and Liu, Q. (2012). An improving deception detection method in computer-mediated communication. *Journal of Networks*, 7(11):1811–1816.

Zhao, Z., Resnick, P., and Mei, Q. (2015). Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1395–1405, Florence, Italy. International World Wide Web Conferences Steering Committee.

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., and Procter, R. (2018a). Detection and resolution of rumours in social media: A survey. *ACM Comput. Surv.*, 51(2):32:1–32:36.

Zubiaga, A., Kochkina, E., Liakata, M., Procter, R., Lukasik, M., Bontcheva, K., Cohn, T., and Augenstein, I. (2018b). Discourse-aware rumour stance classification in social media using sequential classifiers. *Information Processing & Management*, 54:273–290.

Zubiaga, A., Liakata, M., and Procter, R. (2016). Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media. *arXiv e-prints*, page arXiv:1610.07363.

Zubiaga, A., Liakata, M., and Procter, R. (2017). Exploiting context for rumour detection in social media. In Ciampaglia, G. L., Mashhadi, A., and Yasseri, T., editors, *Social Informatics: 9th International Conference*, pages 109–123, Cham. Springer International Publishing.

Zubiaga, A., Liakata, M., Procter, R., Bontcheva, K., and Tolmie, P. (2015). Towards detecting rumours in social media. In *AAAI Workshop: AI for Cities*, pages 35–41.

Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., and Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLOS ONE*, 11(3):1–29.