Università degli Studi di Milano - Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea in Data Science

On the Impact of Different Word Embeddings on Metaphor Detection

Relatore: Prof. Matteo Luigi Palmonari

Correlatore: Dott. Manuel Vimercati

Relazione della prova finale di:
Giorgio Ottolina
Matricola 838017

Anno Accademico 2019-2020

Acknowledgments

Throughout the writing of this dissertation I have received a great deal of support and assistance. I would first like to thank my supervisor, Professor Matteo Luigi Palmonari, whose expertise was invaluable in formulating the research questions and methodology. You gave me the opportunity to work on such a fascinating topic, and your insightful and friendly feedback pushed me to see things from different perspectives, bringing my work to a higher level. I would also like to thank my assistant supervisor Dr. Manuel Vimercati, for his valuable and patient support during these months. You provided me with the tools that I needed to face complex coding problems and complete my dissertation. Above all, you taught me that Google Colaboratory has a Corgi mode. I would like to acknowledge my soon-to-be supervisor during my upcoming traineeship experience at Stockholm University, Professor Ioannis Pavlopoulos, and then Dr. Federico Bianchi and Dr. Mehwish Alam, for their helpful advice and ideas. I would obviously like to thank my parents, Grazia and Carlo, for their continuous support: you have always been there for me and I could not have achieved this goal if it wasn't for you. In addition, I would not have gotten this far without the support of the following people: Marco Ferrario, awesome group mate with whom I completed all the first year projects and survived in Bicocca before I moved to Stockholm and covid-19 completely changed our lives; Federico Mandressi and Alessio Lo Nano, my best friends who have always stood by my side even in the darkest moments of these last years, and whom I know I will always be able to count on, wherever we will be (even if you roast me, I know that deep down you love me). Finally, I have to thank my biggest passion in life and, in some respects, my most intimate friend: music. You are always there for me when I feel the need to creatively express myself, without asking for anything in return. Thanks to you, I have been able to exorcise my demons, find solace even in the most terrifying darkness, and above all, discover who I really am.

Contents

Introduction

Nowadays, with all the technological advancements and computational power for data science tasks at our disposal, natural language processing and analysis is becoming more and more important. Natural Language Processing (NLP) is a wide area of research where the worlds of artificial intelligence, computer science, and linguistics collide. It includes several topics with interesting real-world applications: each one of these topics has its own way of dealing with textual data.

We often think about literal textual data, but not about figurative and metaphorical language, and how difficult it can be for a machine to recognize and understand it. Figurative devices such as metaphors, when used in the right way, offer a fast and flexible (sometimes unconventional) way of conveying meaning from speaker to speaker, or from context to context. Since figurative language is so important in natural language processing, we decided to study this topic through a vast explorative analysis focused on the several main words' representations (embeddings) used in current metaphor detection and identification approaches. Studies have proven that one of the biggest factors that influence language evolution and metaphorical expressions' birth is time. Therefore, a part of our work focused on the analysis of the impact of temporal information on metaphor detection methods, by exploiting temporal word embeddings (such as HistWords - SGNS [136] and Compass-aligned representations [65], that span the last two centuries) inside modified state of the art recurrent neural networks-based models [41] applied on three main literature metaphor datasets (MOH-X [50], VUA [53] and TroFi [35]). These RNN-based models [41] perform metaphor detection as a sequence classification task, by detecting the single metaphorical words inside sentences thanks to mechanisms that take advantage of attention and context. Although the final results reported in the tables 3.7, 3.8 and 3.9 indicate that exploiting temporal embeddings inside the aforementioned recurrent neural networks-based models lead to generally better than state of the art quantitative performances, the overall impact of temporal representations is rather limited. Another interesting pattern that we could observe is the inverse relationship between Precision and Recall scores for VUA and TroFi datasets.

We then expanded our research not only studying these quantitative results, but also taking a qualitative look at the actual metaphorical predictions made by the models using the different embeddings. Thanks to these evaluations, we were able to identify some recurring interesting patterns. For example, topics related to economics, politics and emotions are the most recurring ones in sentences containing correctly identified metaphors; besides, verbs having a literal meaning characterized by physical connotations, often assume figurative meanings when used in sentences related to the previously listed contexts. Finally, studying the predictions obtained with the embeddings of one specific time period, we noticed that no sentences belonging to the 'news' domain of VUA dataset were correctly predicted. This could indicate that for that specific time period (1990 decade), words' representations of that domain are biased towards their metaphorical

meaning, and therefore words are used in metaphorical contexts way more than in literal ones. This would prevent the neural networks from correctly identifying the words as metaphors, since these models generally recognize metaphors because of the mismatch between the words' signals and those of the contexts of the sentences in which they are located.

Another known interesting aspect of metaphors that has also been confirmed by our qualitative analyses, is how they are linked to emotion and feelings. We collected new data (the Language Computer Coorporation dataset [80]) containing both metaphorical sentences and sentiment information, to investigate any new possible patterns highlighted by metaphor detection approaches' predictions. In order to do so, this time we focused on contextual word embeddings, specifically BERT [74], exploiting them within a custom fine-tuned model trained on the three state of the art datasets. The goal was to see whether the models could correctly classify whole metaphorical sentences in the newly acquired test set, while gathering useful information about them, such as links to feelings and emotion, source and target concepts, and so on. Therefore, this time we approached metaphor detection as a binary classification task. Due to time constrainst we were not able to analyze the predictions made on the test set by the fine-tuned model trained on all three state of the art datasets, but the obtained results confirmed the context-related patterns discovered in previous analyses conducted with the other types of embeddings. Emotional and socio-economics topics are the most frequently recurring ones in correctly classified metaphorical sentences.

Taking into consideration these qualitative conclusions and in particular our previously explained hypotheses regarding the possible bias of specific language domains, future research could start from the creation of a new ad hoc dataset based on words with known semantic changes over time. By doing so we would avoid using too many metaphors that have become common figures of speech by now, and we could confirm or deny our hypotheses by checking the new predictions made by the neural networks. Finally, further experimentation with BERT could be performed, since contextual representations are becoming more and more popular and effective in a myriad of current NLP applications.

Chapter 1

Metaphors and Natural Language Processing

1.1 Figurative Language and Metaphors

In this age of the Internet Of Things, where even regular objects can be retrieved and inter-linked via the Web, it is very easy to forget that words were the first inter-linkable things ever. In fact, words are objects as well, with their own forms and specific uses, public identities, private associations and generally adopted contexts. The Belgian surrealist René Magritte famously took advantage of words' duality – we are referring to them being able to refer to objects and to be objects in their own right too – in his provoking exchanging of word and image. In his manifesto on the use of words in pictures, entitled *Les Mots et Les Images* [1], Magritte established a magician's box of tricks for creatively making use of the fragile link between words and what they describe.

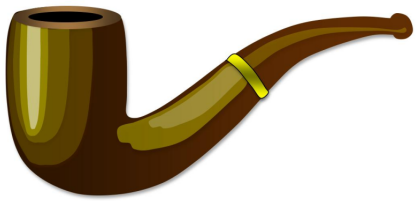

Figure 1.1: René Magritte - 'Ceci n'est pas une pipe'.

The first claim in Magritte's work was particularly important for research and studies about **figurative language**. Words are so useful exactly because they give us the possibility to refer to objects and ideas; anyway, this mapping is neither inevitable nor eternal. As Magritte showed, the linking between words, images and objects is a tenuous one, especially in the hands of a creative person or thinker. In a given context, the most traditionally used word for an object might be the last word we would like to use. At the same time, a word for a totally different tool may actually be adequate for our goals, and communicate our ideas with far stronger energy. Even the philosopher Ludwig Wittgenstein [2] attributed several philosophical problems to words. He used to profess that

philosophers unconsciously create ambiguity and confusion by 'taking words on holiday': words are extracted from their traditional contexts, where they make intuitive sense, and exported to more exotic contexts where their meanings are stretched. This is what we do when we deliberately adopt language in a creative way: by taking our words on holiday, we allow them to abandon their conventional contexts so that they can show us exciting and new sides of themselves. Figurative devices such as **metaphors**, when used in the correct way, offer flexible (most of the times unconventional) and fast means of conveying meaning from speaker to speaker, or from context to context. If they are used in the wrong way though, meanings become like lost luggage: misplaced and irretrievable, with their contents often dangerously misunderstood. We adopt words in a figurative way to summon other meanings and words and to bring them into play: Magritte captured this consideration in his surrealist manifesto, by saying: *A word can replace an object in reality* and *An object makes one suppose there are other objects behind it.* It is surely advantageous to think of metaphors as performing a unidirectional information transfer, where knowledge gets transferred from a source concept, and projected onto a target one, but this information flow can be bidirectional at the same time. Let's have a look at this old remark about media guru Arianne Huffington, the founder of news blog 'The Huffington Post': *the most upwardly mobile Greek since Icarus.* [3] We can see how Huffington and Icarus are both described as social climbers in this metaphor, compelling us to update our personal ideas regarding them: now Icarus is seen as a social climber of sorts too. Arianne Huffington and the Greek mythological figure are actually blended together into one ambitious person thanks to this metaphor. The conclusion is that figurative language exploits the way in which our conceptual structures are structured, and even allows us to rewire them, by connecting seemingly unrelated perceptions in interesting new ways. Just like with computer cables, that generally come in different sizes and bandwidths, a figurative link is able to carry a single piece of important information or, in parallel, a large amount of related information. For example, a scientific analogy is capable to create a whole system of coherent mappings between source and target domains, while a humorous kind of analogy might build an incredibly detailed source picture to convey just one small piece of knowledge. Finally, let's consider this Shakespeare's oft-quoted metaphor from Romeo and Juliet: "Juliet is the sun." [4]. This metaphor gives away more than the simple brilliance of Juliet's beauty, as perceived by her lover Romeo. The metaphor also conveys a bigger system of metaphoric mappings that runs throughout the entire play. In this metaphorical solar system, Juliet is considered as the gravitational center around which all the other characters continuously orbit. This metaphor can be seen as a thin-pipe that delivers a single piece of information, but at the same time as a fat-pipe which conveys the richness of a solar system related metaphor. Many creative and humorous figures and analogies force us to build a complex image of the source-domain mapping. In this way, a significant quality gets inferred from a mental image, and it is projected into the target domain. **Metaphors, similes, analogies** and other figurative figures are knowledge-hungry devices. Creative language takes advantage of very different types of knowledge:

1. Propositional knowledge (events, actions, behaviors, norms);
2. Property-level knowledge (expectations, category membership criteria);
3. Semantic knowledge, such as dictionaries, as opposed to pragmatic knowledge such as corpora.

1.2 Metaphors in NLP

According to the *Conceptual Metaphor Theory (CMT)* proposed by Lakoff and Johnson [5], a metaphor:

1. is not just a property of language, but rather a cognitive mechanism that structures our conceptual system;

2. involves a complex cross-domain knowledge projection process.

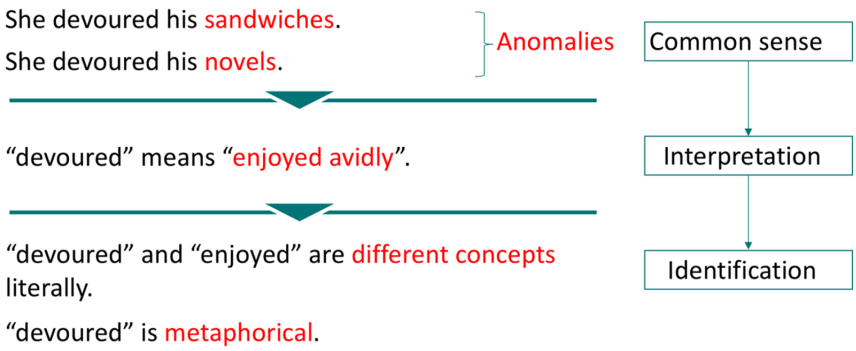

Figure 1.2: Metaphorical expressions are usually made more easily identifiable by context.

> *"The wheels of Stalin's regime were well-oiled and already turning,"*
>
> Lakoff and Johnson, 1980: 'Metaphors we live by' [5]

This metaphorical sentence by George Lakoff and Mark Johnson helps us understand that a metaphor manifests itself through a presence of a mapping between two domains of experience: the target (in our case, politics) and the source (in our case, mechanism). Metaphors arise due to systematic association between distinct and seemingly unrelated concepts. The existence of this association allows us to transfer knowledge and inferences from the domain of mechanisms and discuss them using the mechanism terminology. The semantic distance between 2 concepts is, according to Lakoff [5], what makes a metaphor *more metaphorical* than another one. Wilks, in his work *Making Preferences more active* [8], which is considered the most influential one regarding metaphor recognition for automatic metaphor recognition in text, affirms that metaphors are 'a violation of semantic constraints put by verbs onto their arguments'. Fass' approach (*Processing Metonymy and Metaphor* [6] was one of the first real approaches to metaphor detection, and it consisted in the distinction between **literalness**, **metonymy**, **metaphor**, and **anomaly**: phrases could be tested for being a metonymic relation using hand-coded patterns (e.g.: container-for-content). One of the biggest problems with this approach is that interpretation is always context-dependent. A phrase could either be metaphorical or literal, for example: *All men are animals*. Peters et al. [9] made use of *WordNet* (a large lexical database of English nouns, verbs, adjectives and adverbs grouped into sets of cognitive synonyms called synsets, each expressing a distinct concept) hierarchy to group close

senses and to find hyponymy relations. The semantic field of a hyponym is included within that of its hyperonym, for example: *spoon* is a hyponym of *cutlery*. If two words are not included in WordNet's hyponymy/hypernymy hierarchy, then they are most likely part of a metaphorical phrase. CorMet was the first system ever to automatically discover source-target domain mappings. Following Fass' work [6], state of the art approaches to metaphor detection have begun spreading. Shutova's survey [7], *Design and Evaluation of Metaphor Processing Systems*, provides a detailed look of the various methods.

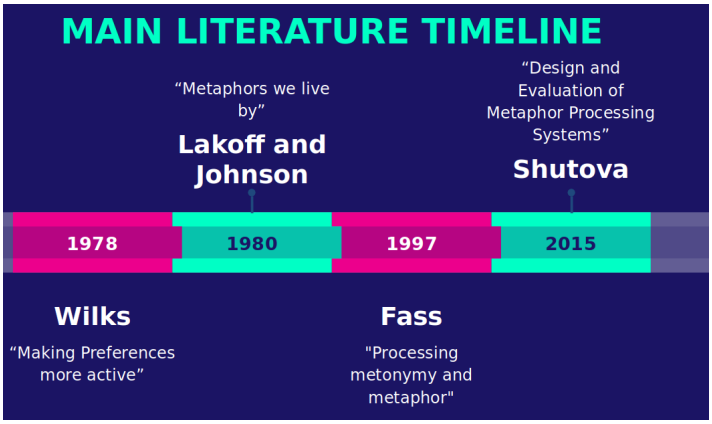

Figure 1.3: Main metaphor-related studies' timeline

Metaphor annotation studies have typically been corpus-based and they involved either continuous annotation of metaphorical language (i.e., distinguishing between literal and metaphorical uses of words in a given text), or search for instances of a specific metaphor in a corpus and an analysis thereof. Metaphor use in everyday language is a way to relate our physical and familiar social experiences to a multitude of other subjects and contexts (Lakoff and Johnson, [5]); it is a fundamental way to structure our understanding of the world, even without our conscious realization of its presence as we speak and write. It highlights the unknown using the known, explains the complex using the simple, and helps us to emphasize the relevant aspects of meaning resulting in effective communication. We can see a few examples in figure 1.4.

1.2.1 Metaphor Detection in Real World Applications

Natural Language Processing is a branch of data science that consists of systematic processes for analyzing, understanding, and deriving information from text data. By using the techniques of Natural Language Processing, one can organize and analyze the massive chunks of text data, perform numerous automated tasks to solve a wide range of problems. Metaphor detection is not only used in several main NLP practical applications, but it is also crucial to maximize their performances, if exploited correctly. Some of the most important NLP applications where metaphor detection is implemented are the following:

M: *The alligator's teeth are like white* ***daggers***
I: The alligator have white and pointed teeth.

M: *He* ***swam*** *in a* ***sea*** *of diamonds.*
I: He is a rich person.

M: *Authority is a* ***chair****, it needs* ***legs*** *to* ***stand****.*
I: Authority is useless when it lacks support.

M: *In Washington, people change* ***dance partners*** *frequently, but not the* ***dance****.*
I: In Washington, people work with one another opportunistically.

M: *Robert Muller is like a* ***bulldog*** *— he will get what he wants.*
I: Robert Muller will work in a determined and aggressive manner to get what he wants.

Figure 1.4: Metaphorical sentences (M) with their relative metaphors in bold, and their literal versions (I)

1. **Machine Translation**: A machine translation system uses natural language processing techniques to build systems that are capable of automatic language translation. If we read a post in another language on Facebook and see its translation just below it, or if we open a website of any language other than our own in Chrome browser or even use Google Translate on a trip to a foreign country, then we are using some kind of a machine translation system;

2. **Text Summarization**: In today's busy world, people need byte sized summaries of information to effectively take action on it without indulging time more than necessary. Automatic Text Summarization is one of the most interesting problems in the field of NLP. It is a process of generating a concise, coherent and meaningful summary of text from text resources such as books, news articles, blog posts, research papers, etc. Automatic text summarization can broadly be divided into two categories: *extractive summarization* and *abstract summarization.* In the first aforementioned approach, several parts from the original text document are extracted, such as phrases and sentences, from a piece of text and stacked together to create a summary. In the second one, advanced NLP techniques are used to generate an entirely new summary. Some parts of this summary may not even appear in the original text;

3. **Question Answering**: Chatbots are everywhere today, from booking our flight tickets to ordering food. Customers nowadays don't want to wait for hours just to get their queries resolved, and they demand instant answers. We currently have many conversational agents or AI assistants like Alexa, Siri, Cortana and Google Home, that all use natural language processing internally;

4. **Opinion Mining**: Opinion mining, or sentiment analysis, is a text analysis technique that uses computational linguistics and natural language processing to automatically identify and extract sentiment or opinion from within text (positive, negative, neutral, etc.). It can allow a business to find out what customers like and

dislike, and why, in order to create products and services that meet their needs. With the right tools, opinion mining can be performed automatically, on almost any form of unstructured text, with very little human input needed. Sentiment analysis can process thousands of pages, comments, emails, or surveys in just minutes for real-time results.

1.2.2 IS-A & OF Metaphors

The main types of semantic features for IS-A & OF metaphors can be summarized as follows: Leacock-Chodorow, Resnik, Wu-Palmer, Jiang-Conrath, Lin, Path Distance Similarity (similarity measures present in WordNet); other similarity measures adopted using Google's search engine, such as normalized Google distance and point wise mutual information; concreteness measures adopted using WordNet. The identification of this kind of metaphors, as explained by Bogdan and Costin in *Metaphor Detection* ([40]), is divided in several steps:

1. Initially, a corpus of examples is taken (in Bogdan and Costin's work ([40]), specifically, the corpus was extracted from the **Master Metaphor List**;
2. examples containing possible forms of the verb *be* are manually or automatically extracted;
3. examples that actually contain *be* forms are now manually or automatically extracted;
4. the previously extracted examples get tagged as either metaphorical or not;
5. named entities are transformed into their categories (*Socrates*, for example, becomes *the person*;
6. the resulting nouns are lemmatized.

If hyponym-hyperonym relation is found in WordNet between two nouns, the observed sentence is literal, otherwise it is metaphorical. In linguistics and lexicography, a *hypernym* is a word whose meaning includes the meanings of other words. For instance, flower is a hypernym of daisy and rose. Hypernyms (also called superordinates and supertypes) are basically general words; *hyponyms* (also called subordinates) are subdivisions of more general words. The semantic relationship between each of the more specific words (e.g., daisy and rose) and the more general term (flower) is called hyponymy or inclusion ([131]).

1.2.3 Verb Metaphors

Always referring to Bogdan and Costin's work ([40]), verb metaphors are mainly based on text categorization features: the most common type of these features are bag of words ones. Low information features generally have to be removed in order to increase performance and to reduce over-fitting (a classifier gets a very good performance on a training set, but does not have the capability of generalizing well on other unseen datasets). The methods used to modify the examined features are mainly document frequency thresholding, chi-statistic, term strength, information gain and mutual information.

1.2.4 Metaphors and Emotion

Metaphors do way more than delivering propositions: they convey feelings regarding those propositions. Successful metaphors are the common currency of a language. They are used everyday, with well-understood meanings, feelings and communicative functions. The most common metaphors are so conventionalized that it is indeed difficult to recognize them as such. A landmark analysis of conventional metaphors is provided in George Lakoff and Mark Johnson's book *Metaphors We Live By* ([5]): these metaphors are incredibly pervasive and cognitively entrenched in our language. How could a computer, which doesn't possess emotions of its own, appreciate the feelings that a metaphor can summon? What are for example the emotional echoes of a property like *bloody*? And what are instead the echoes of a concept that is stereotypically bloody, such as a murderer? Similes can be used as a guide: what kind of feeling does bloody evoke? The obvious answers are those properties p that can be expressed in the following manner: *I feel p-ed by ....* Sentiment analysis' aim is to detect the affective attitude in text. A vast majority of work in sentiment analysis has focused on developing classifiers for valence prediction (Kiritchenko et al., [68]; Socher et al., [69]; Mohammad et al., [70]), i.e., determining whether a piece of text expresses positive, negative, or neutral attitude. There is also a growing interest in detecting a wider range of emotions such as joy, sadness, optimism, etc. (Holzman and Pottenger, [71]; Brooks et al., [72]; Mohammad, [73]). In *Metaphor as a Medium for Emotion: An Empirical Study* by M. Mohammad, Ekaterina Shutova et al. ([50]), two questions are addressed:

1. Whether a metaphorical statement is likely to convey a stronger emotional content than its literal counterpart;

2. How this emotional content is born in the metaphor, for example if it generates from the source domain, or from the target domain, or rather arises through the source and the target's interaction.

To answer these questions, a series of experiments are conducted by the authors, in which human subjects are asked to evaluate metaphoricity and emotionality of a sentence in several settings. Two experimental hypotheses are then tested:

1. Hypothesis 1: metaphorical uses of words tend to deliver more emotion than their literal paraphrases in the same context;

2. Hypothesis 2: the metaphorical sense of a word tends to bring with itself more emotion than the literal sense of the same word.

The results supported both hypotheses, providing evidence that metaphor is an important mechanism for expressing emotions. Besides, the fact that metaphorical uses of words carry more emotion than their literal uses suggests that the emotional content is not just transferred from the source domain into the target, but it is rather a result of the interaction between the two domains in the metaphor. Are there specific sentiments or feelings that make metaphorical sentences easier to be correctly recognized as such then? We will investigate this matter in Chapters 3 and 4.

Chapter 2

State of the Art

2.1 Machine Learning

Machine learning is the study of computer algorithms that improve automatically through experience ([144]). It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as 'training data', in order to make predictions or decisions without being explicitly programmed to do so. ([145]). There are two main types of supervised learning problems: **classification** or **regression** problems. Classification problems' goal is to classify data into a finite number of categories, while regression problems aim to predict a continuous number as their output. Machine learning problems are called supervised learning problems if the dataset also contains the true labels to compare predictions against. For example, in our case, based on a text sentence x_i , a learning model H should classify the metaphorical nature y_i in the sentence. The predicted metaphorical labels $\hat{y}_i$ is then compared to the actual metaphorical ones to evaluate our model. In machine learning, *multi-label classification* and the strongly related problem of multi-output classification are variants of the classification problem where multiple labels may be assigned to each instance. Multi-label classification is a generalization of multi-class classification, which is the *single-label* problem of categorizing instances into precisely one of more than two classes; in the multi-label problem there is no constraint on how many of the classes the instance can be assigned to ([153]). In this work, we will focus on the following areas and approaches:

1. **Deep Learning**: it is a sub-field of machine learning concerned with algorithms inspired by the structure and function of the brain: these algorithms are called artificial neural networks, and we will soon analyze them in detail. Deep learning was first theorized in the 1980s, when parallel distributed processing became popular under the name connectionism. Rumelhart and McClelland (1986) described the use of connectionism to simulate neural processes ([146]). Deep learning is mainly used because of these reasons: it requires large amounts of labeled data (e.g.: driver-less car development needs millions of images and thousands of hours of video), and it needs substantial computing power to work. GPUs are essential to run deep learning tasks because of their high performances; besides, they are easy to use online thanks to their parallel architecture. Thanks to clusters and cloud computing, development teams can reduce training time for a deep learning network from weeks to hours or even less. One of the main differences between machine learning and deep learning is that the latter performs *end-to-end learning* ([149]): a network is given raw

data and a task to perform, such as classification, and it learns how to do this automatically;

2. **Transfer learning**: is a machine learning method where a model developed for a task is reused as the starting point for a model on a second related task. ([130]) Given the enormous resources and large datasets that are used nowadays, it is not feasible to re-train models from scratch every time. Besides, the huge dimensions of the datasets make the re-trained models capable of capturing generic information, which can be useful for different kinds of tasks as well. It is a popular approach in deep learning where pre-trained models are used as the starting point, for example on NLP tasks. Transfer learning is related to problems such as multi-task learning and concept drift and it is not exclusively a deep learning area of study. Nevertheless, transfer learning is popular in deep learning given the enormous resources required to train deep learning models or the large datasets on which deep learning models are trained. This form of transfer learning used in deep learning is called inductive transfer ([147]): the scope of possible models is narrowed in a beneficial way by using a model fitted on a different but related task. Transfer learning is commonly used with natural language processing problems that use text as input or output ([148]).

Some of the main approaches used in literature to perform metaphor detection are:

1. **Logistic Regression**: a model performs binary classification, so the label outputs are binary. Logistic regression provides several ways to regularize models and to adjust features' correlation, alongside easy probabilistic interpretation and lots of ways to update models in order to take in new data. In NLP, this algorithm can be used for example to understand whether comments are positive or negative;

2. **Linear Regression**: Regression algorithms can be used when the goal is to compute continuous values, whereas classification models' output is categorical. Regression is generally used when some future value of a process which is currently running has to be predicted (e.g.: predicting sales of a particular product during the next month). One real world case where linear regression algorithm can be applied to natural language processing (NLP) is rating prediction based on review text;

3. **Decision Trees and Random Forests**: Decision trees easily handle feature interactions and they are non-parametric, therefore there is basically no need to worry about outliers or whether the data is linearly separable ([149]). One disadvantage is that these models don't support online learning, so trees need to be rebuilt when new examples come on ([149]). Another disadvantage is that they easily overfit and take a lot of memory to work, but that's where ensemble methods like random forests or boosted trees come into play ([149]). Random Forests are ensembles of decision trees. They can solve both regression and classification problems with large data sets. They also help identify most significant variables from thousands of input variables. Random Forests are highly scalable to any number of dimensions and generally achieve quite acceptable performances ([149]);

4. **Naive Bayes**: it is a classification technique based on Bayes' theorem ([132]), easy to build and particularly useful for very large data sets. Naive Bayes is also known

to outperform even highly sophisticated classification methods, and it is a good choice when CPU and memory resources are a limiting factor. If the Naive Bayes conditional independence assumption ([150]) actually holds, this kind of classifier will converge quicker than models like logistic regression, so there is no need for huge quantities of training data ([151]). Naive Bayes' main disadvantage is that it can't learn interactions between features. In NLP, this algorithm can be used for sentiment analysis and text classification, recommendation systems like Netflix or Amazon, and to mark emails as spam or not spam;

5. **Support Vector Machine (SVM)**: it is a supervised machine learning technique that is widely used in pattern recognition and classification problems. SVM models can achieve high accuracy and, with an appropriate kernel, they can work well even if the data is not linearly separable in the base feature space ([152]). They are especially popular in text classification problems where very high-dimensional spaces are common. SVMs are however memory-intensive, generally hard to interpret, and difficult to tune;

6. **Clustering**: it consists in dividing the population or data points into a number of groups such that the data points in the same groups are more similar to other data points in the same group than those in other groups. In simpler terms, the goal of these models is to separate groups with similar traits and assign them into clusters. Clustering can be divided into two sub-groups: hard clustering, where each data point either completely belongs to a cluster or not, and soft clustering, where instead of putting each data point into a separate cluster, a probability (or likelihood) of that data point being in those clusters is assigned. In NLP, clustering algorithm is often used in recommendation systems and social network analysis.

2.1.1 Classification Models' Evaluation

This work is focused on classification models. There are several ways of measuring how well a classification model performs in predicting the data. Accuracy, defined as the percentage of correctly classified samples, is possibly the most intuitive measure, but not the most useful one. In fact, in a scenario where the class sizes are uneven, then the largest class will dominate accuracy and the model could get a pretty good accuracy score just by correctly predicting the largest class every time. Therefore, in a multi-class setting, models' performances are measured against other metrics such as Precision, Recall and the F1 score. When a model predicts that a data example is a member of a specific class for a given example, it is by default predicting that this data is not a member of the other $k-1$ classes. We will sometimes use a **confusion matrix** to have a more complete picture when assessing the performance of a model. A confusion matrix is defined as follows (see figure 2.1):

A **true positive** is an outcome where the model correctly predicts the positive class. Similarly, a **true negative** is an outcome where the model correctly predicts the negative class. A **false positive** is an outcome where the model incorrectly predicts the positive class. Similarly, a **false negative** is an outcome where the model incorrectly predicts the negative class. To summarize, the principal evaluation metrics for machine learning models are the following:

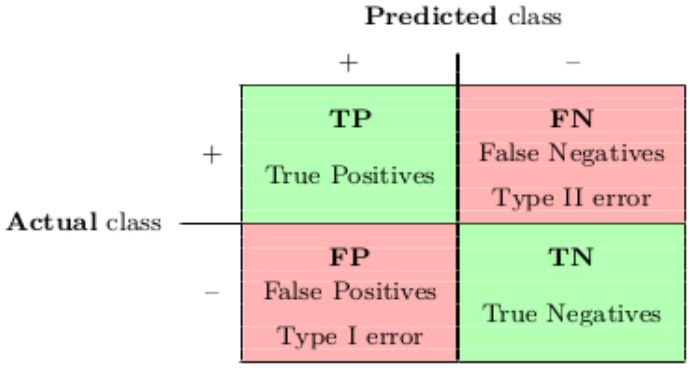

Figure 2.1: An example of a confusion matrix.

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Accuracy is the overall performance of the model.

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

Precision provides information about how accurate the positive predictions are.

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

Recall is the coverage of actual positive sample.

$$\textbf{F1 score} = \left(1 + \beta^2\right) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{2.4}$$

F1 score is a hybrid metric useful for unbalanced classes: it is the harmonic mean of precision and recall.

2.1.2 Artificial Neural Networks

Artificial Neural Networks (ANNs), first proposed by Mc-Culloch and Pitts (1943) [87], are one of the fundamental building blocks of machine learning. ANNS are based on the structure of neurons in the brain. These structures were proven to be applicable to a wide range of problems when combined through different architectures. We first begin with the representation of data. In figure 2.2, we can see a p-dimensional dataset X of which we have n samples:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ \vdots & \ddots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

Figure 2.2: Representation of data for artificial neural networks.

Here, each row is a vector x_i where $i = 1, ..., n$ has a corresponding label denoted y_i. This output vector has to be modeled as well as possible based on the data at our disposal. The simplest way to model the problem is through the usage of a linear combination of the input, performing a linear regression. This leads to the linear regression model shown in equation 2.5.

$$\hat{\mathbf{y}} = V\mathbf{X}^\top \tag{2.5}$$

Where each element of $V \in R^p$ is a real number called a *weight*. Instead of only outputting one single value, it is possible to create a model which outputs a vector of K values by increasing the dimension of the original weight vector. From there, we can use the *softmax function* to scale these outputs to probabilities between 0 and 1, representing the probability that the output belongs to that specific class. Softmax ϕ is computed as shown in equation 2.6:

$$\phi(\mathbf{z})_j = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)} \tag{2.6}$$

For $j = 1, ..., K$, where K is the number of classes. We can construct predictions by letting V be a matrix of dimensions *($K \times p$)*. Then we obtain the model presented in 2.7:

$$\hat{\mathbf{y}}_i = \phi\left(V\mathbf{x}_i^\top\right) \tag{2.7}$$

Then the $\hat{\mathrm{y}}_i : K$ becomes the predicted probabilities that the output label belongs to each class. In equation 2.5, we saw how to formulate the task in a completely linear setting. Then, by adding a non-linear function (e.g.: softmax), we saw that we can model more complex behavior. However, the real strength of ANNs come when we create compositions of non-linear functions that all contain weight parameters. This combination of functions and weight parameters is where we enter the domain of deep learning, where the y_hat value is computed by a composition of functions, each function h is computed by a hidden layer and is given by equation 2.8:

$$\mathrm{h} = \sigma\left(U\mathrm{X}^\mathrm{T}\right) \tag{2.8}$$

The elements of the matrix U are also called weights and the function *sigma* is referred to as the hidden layer's activation function. For classification purposes, softmax is a good choice for the output layer. Non-linear behavior is often desirable for activation functions, therefore popular activation functions include the rectified linear unit (RELU) (Hahnloser et al., 2000) [88] and hyperbolic tangent (tanh) among others.

2.1.3 Recurrent Neural Networks

Language can originally be thought as a sequence, since we read words in order. Recurrent neural networks (RNNs), proposed by Elman (1990) [89] among others, take into account the sequential nature of the input when making output predictions: thanks to this method, important language information about which words come in which order in the sequence are not lost. RNNs accomplish this by computing a set of feedback weights in a hidden state vector at each time step in the sequence that pass information from

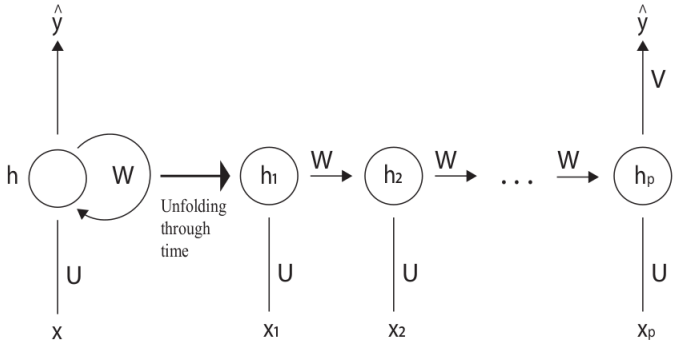

Figure 2.3: Recurrent neural network schema.

earlier time points. In order to predict whether a sequence belongs to a specific class, the model can be reformulated to take into account this new time dependency by using the final recurrent hidden state vector, thus making softmax probability predictions.

$$\hat{y} = \phi\left(V\mathbf{h}_p\right)\ c \tag{2.9}$$

Where

$$\mathbf{h}_p = \sigma\left(U\mathbf{x}_p + W\mathbf{h}_{p-1}\right) \tag{2.10}$$

The model's parametrization ω now gets comprised of the weight-matrices U, V and W. The hidden layer is now dependent on earlier states.

2.1.4 Long Short-Term Memory

(Hochreiter and Schmidhuber, 1997) [90] and (Gers et al., 2000) [91] showed how adding gating to RNNs allows them to learn long-term dependencies. A memory of long sequences is obtained by adding elements to each recurrent layer. The short-term memory capabilities are unchanged if compared to the simple RNN, leading to the long short-term memory unit (LSTM). LSTM networks contain two main innovations from the simple RNNs. Firstly, at each time step a hidden state vector and a local context vector are passed to the next recurrent node. Secondly, the LSTM network consists of a set of gating mechanisms that enables the model to decide which kind of information to pass forward in recurrence. The LSTM model can steadily learn long-term dependencies in the sequences. The gating mechanisms are broadly defined as an input gate, a context gate, a so-called forgetting gate and an output gate. These gates, and their related weights, are defined by the following relationships:

$$\mathbf{f}_p = \sigma\left(\mathbf{x}_p U^f + \mathbf{h}_{p-1} W^f\right) \tag{2.11}$$

$$\mathbf{i}_p = \sigma\left(\mathbf{x}_p U^i + \mathbf{h}_{p-1} W^i\right) \tag{2.12}$$

$$\mathbf{o}_p = \sigma\left(\mathbf{x}_p U^o + \mathbf{h}_{p-1} W^o\right) \tag{2.13}$$

These functions are all recurrent on the previous time step's hidden state $\mathbf{h}_{p-1}$ and the current input data at the time period $\mathbf{x}_p$.

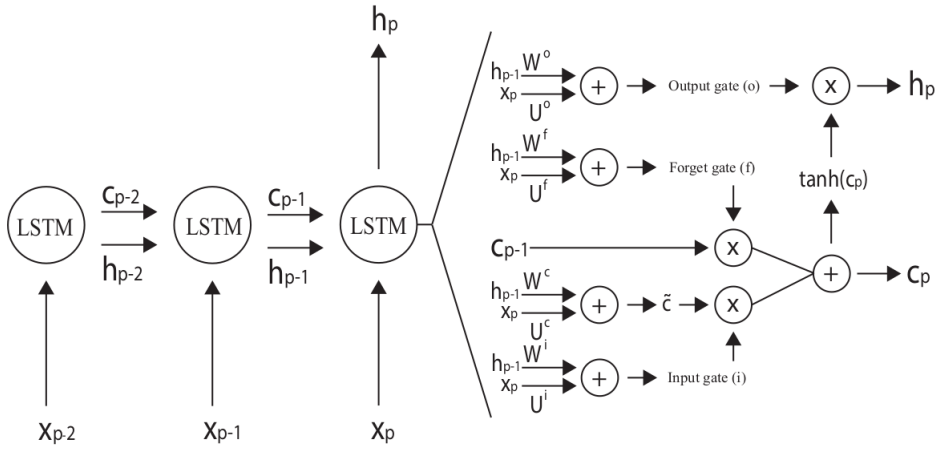

Figure 2.4: Long short-term memory network. To the right, the inner workings of a node is shown.

2.1.5 Bidirectionality

Not every word is predictable based on the word before. If we consider the sequences "river bank" and "bank account", *bank* has a contextual dependency in the two sequences, not only on the previous word, but on the word after as well. Regular recurrent networks can struggle with this problem, which can be solved however thanks to **bidirectionality**, proposed by Graves and Schmidhuber (2005). [92] This means reversing the direction of the sequence and feeding it to an independent network by concatenating the resulting hidden states. For many language models, this has currently become the standard practice ([129]). In figure 2.5 we can see a Recurrent Neural Network that works bidirectionally:

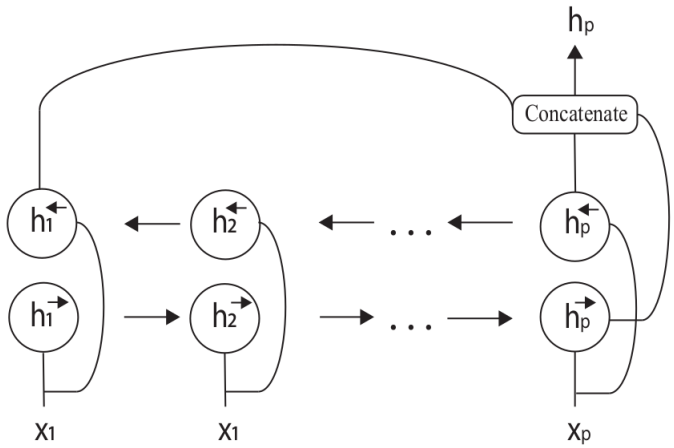

Figure 2.5: Bi-directional recurrent neural network.

The states ($\mathbf{h}_p$) in figure 2.5 might be from regular recurrent nodes or LSTMs. The network with the reversed states is an independent copy of the original network. Commonly, the resulting hidden states are concatenated to form:

$$\mathbf{h}_p^{\rightarrow} = (\mathbf{h}_p, \mathbf{h}_1^{\leftarrow}) \tag{2.14}$$

This last hidden state vector can then be utilized in the same way as for simple Recurrent Neural Networks. Since extra contextual information is added to language models through bidirectionality, this method gets often employed to improve model representation and predictions.

2.2 Word Embeddings

In natural language processing, given a vocabulary composed of N words, a one-hot vector is a $1 \times N$ matrix utilized to distinguish each word in a vocabulary from every other word in the same vocabulary. One-hot encoding leads to mathematically independent vectors' representations. The one-hot vector consists of 0s in all cells with the exception of a single 1 in a cell used just to identify the word. One-hot encoding ensures that machine learning does not assume higher numbers as being more important. For example, the value '8' is bigger than the value '1', but that does not make '8' more important than '1' ([127]). Simple one hot vectors are not a very useful input for most NLP tasks, because they are embedded in a vector space that does not contain any extra meaning information about the words being represented. Besides, as vocabulary size grows, one hot encoded vectors become computationally unusable since each word needs its own separate dimension, so that $x_i \in R^N$. A more efficient method consists in passing a lower dimensional vector $x_i \in R^N$ $(d < N)$, which embeds language information about the word w_i too. The newly obtained language information is assumed to be more relevant than that of a one-hot encoded vector. This type of lower dimensional representation is called a **word embedding**, since it embeds words in a multi-dimensional meaning space where groups of semantically and syntactically similar words are located near to each other in terms of vector distance. Traditional word embeddings are generally based on a classic dimension reduction technique called *principal component analysis (PCA)* ([128]), which relies on the singular value decomposition of the co-occurrence matrix. This approach is unsupervised and thus particularly advantageous, since no specific linguistic rules need to be fed to the system. Besides, PCA looks globally at the text corpus for all of the occurrences of a word in all the different contexts, leading rich language features. This technique can easily become too computationally expensive. This problem can be solved in an unsupervised way, through neural techniques as shown by Bengio et al. (2003). [95] A feed forward neural network is used to predict the *n-gram* probabilities of the words in the vocabulary given the context that came before them. For each word in order in the text corpus, the n previous words are utilized as context, and each word gets represented by a trainable, d-dimensional random initialized vector. These vectors are then concatenated and fed through several hidden layers. At this point, the output layer becomes a softmax probability over the vocabulary of all the probabilities of seeing each word, given the context words. This objective function ensures that the network is learning linguistically valuable information in the matrix of d-dimensional vectors while training to predict n-grams. It was shown by Collobert et al. (2011) [96] that this type of word embedding is not just a placeholder for words, but actually coded important

language information. Word embeddings can indeed be used as the only input to other machine learning models for applied natural language processing tasks with state of the art results. Word embeddings were also proved to be useful in finding named entities in phrases (Turian et al., 2010) [97], and they can be aligned over languages to help machine translation (Zou et al., 2013) [98]. Bengio's model ([95]), however, had several drawbacks. Firstly, it was based on a feed forward neural network (figure 2.6), therefore the only way possible to increase the network's context windows that has to be learned was to increase the number of input nodes. This lead the embeddings to be expensive to train for recognizing context-heavy features inside the data, such as emotional information.

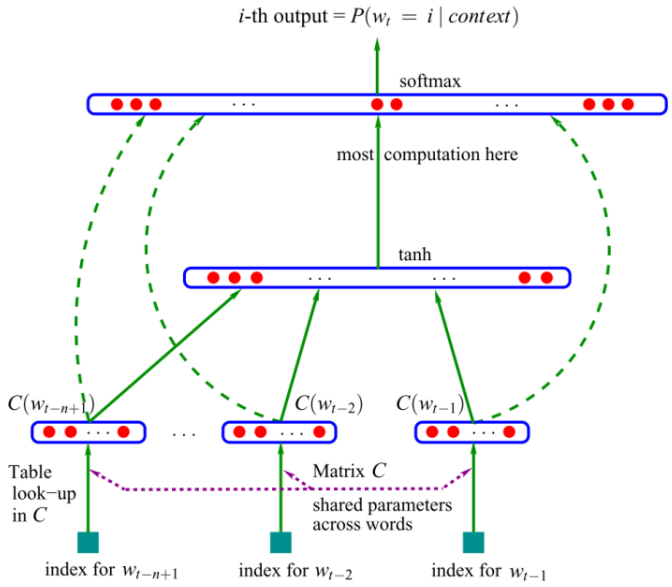

Figure 2.6: Feed forward neural network.

Unsupervised word embeddings methods have been proved to be particularly useful at identifying and interpreting metaphors at word-level with very little or any preprocessing, outperforming strong baselines in the metaphor identification task.

2.2.1 Static Word Embeddings

Word2Vec

The Word2Vec model was created by Google in 2013 ([93] - [94]) and is a predictive deep learning based model to compute and generate high quality, distributed and continuous dense vector representations of words, which capture contextual and semantic similarity. Word2Vec model elaborates massive textual corpora, and generates dense word embeddings for each word in the vector space representing that vocabulary, using a vocabulary of possible words as input. It is usually possible to specify the size of the word embedding vectors and the total number of vectors, that are basically the size of the vocabulary. There are two different model architectures which can be leveraged by Word2Vec to create these word embedding representations:

1. **The Skip-gram Model**: In order to train more efficiently on smaller datasets and with better context information, Mikolov et al. (2013 [93] - [94]) proposed an embedding creation approach called the *skip-gram* model (see figure 2.7). The output of the skip-gram model (2.7), instead, is the probability of different context words, based on a target word. This technique embeds word-vectors in a universe of language meaning which is very context rich and semantically interpretable, especially in terms of relations between words. By using vector arithmetic, for example, the vector for *King* minus the vector for *Man* is extremely close in vector space to the vector for *Queen*: this shows that the model successfully learns to represent words with respect to a dimension which we can understand as 'gender', and another one that we can understand as 'royalty'.

2. **The Continuous Bag of Words (CBOW) Model**: The CBOW model (see figure 2.8) architecture tries to predict the current target word based on the source context words (the surrounding words). Looking at a sample sentence such as "the quick brown fox jumps over the lazy dog", this can be considered as pairs of *(context_window, target_word)* where if we consider a context window of size 2, we have examples like '([quick, fox], brown)', '([the, brown], quick)', '([the, dog], lazy)', and so on. Therefore the CBOW model tries to predict the target_word based on the context_window words. While the Word2Vec family of models are unsupervised (it is possible to just give it a corpus without additional labels or information and dense word embeddings will be constructed from the corpus), a supervised classification methodology is still needed to get to the embeddings from the corpus. The CBOW architecture can be modeled as a deep learning classification model where the context words are taken as the input X in order to predict the target word Y. This architecture is simpler to build than the skip-gram model, where the aim is to predict a whole bunch of context words from a source target word.

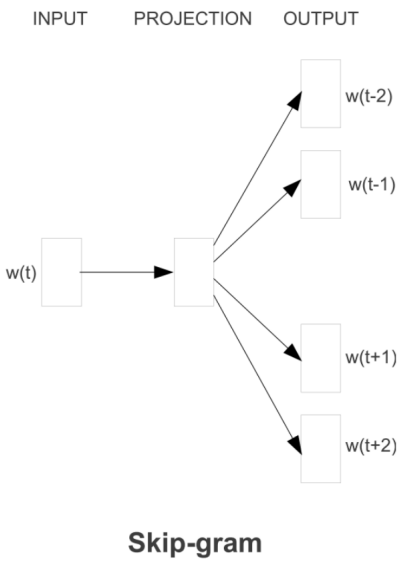

Figure 2.7: Skip-gram, feed forward word embedding architecture.

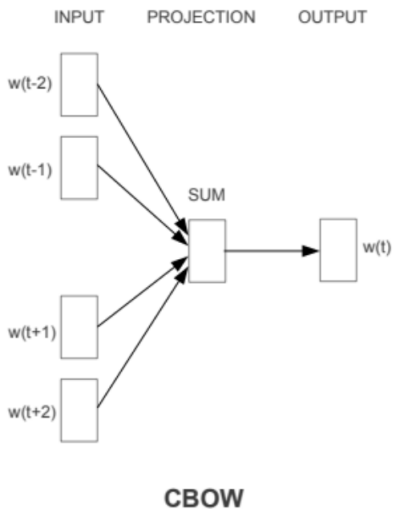

Figure 2.8: CBOW word embedding architecture.

GloVe

GloVe (Global Vectors) embedding, from Pennington et al. (2014) [45], is different from the skip-gram and feed-forward word embedding models because it is trained on a loss function which takes into account both local co-occurrences from the n-length context windows, but also global count-based co-occurrence probabilities from the text corpus. This is an attempt to encode more language features than it would be possible with a technique like PCA, and it is based on the intuition that including only local context information does not give features enough information regarding how often words occur in rare contexts. The loss function ends up looking similar to the skip-gram model's one, but with slightly richer context specific information embedded in the vectors. The original GloVe model [45] is trained on five corpora of various dimensions:

1. a 2010 Wikipedia dump with 1 billion tokens;
2. a 2014 Wikipedia dump with 1.6 billion tokens;
3. Gigaword 5 with 4.3 billion tokens;
4. Gigaword5 + Wikipedia2014, with 6 billion tokens in total;
5. Common Crawl, which is 42 billion tokens of web data

Other Embeddings

Some other popular embeddings models include the FastText embeddings (Joulin et al., 2016) [99] and the ConceptNet embeddings from Speer et al. (2016) [100].

2.2.2 Contextual Word Embeddings

Static embeddings do not change with the context once been learned. Despite their efficiency, the static nature of these embeddings makes it difficult to cope with the polysemy

problem, since the meaning of a polysemous word depends on its context ([154]). To deal with this problem, a number of approaches have been recently proposed to learn the representation of words among their contexts. For example, in two sentences: "Apple sells phones" and "I eat an apple", dynamic embeddings will represent "apple" differently according to the contexts, while static embedding can not distinguish the semantic difference between two "apples" ([154]). These dynamic embeddings extracted from pre-trained language models ([155], [156], [157], [158]) have demonstrated dramatic superiority over their static predecessors in various NLP tasks. The main contextual word embeddings are **ELMO** ([129]) and **BERT**. The latter is based on the Hugging Face **Transformers** architecture introduced by Vaswani et al., [49].

ELMO

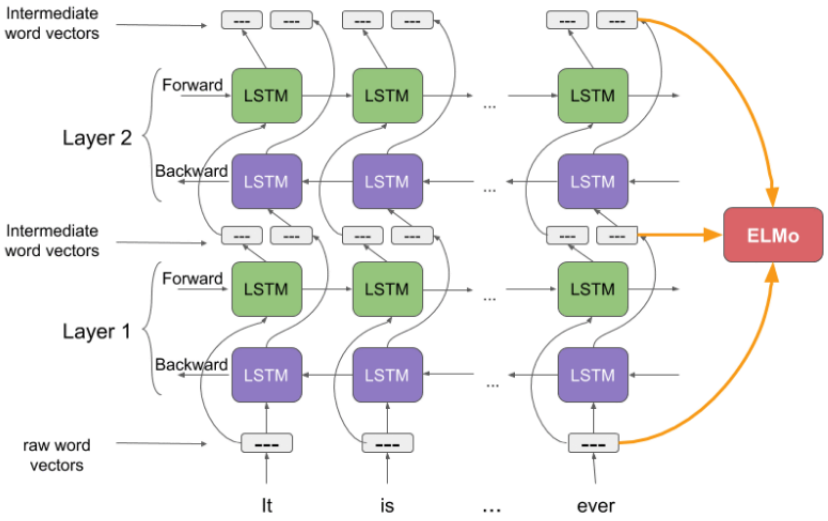

Figure 2.9: Visual representation of how ELMO works.

ELMo ([129]) word vectors are computed on top of a two-layer bidirectional language model (*biLM*). This particular model consists of two layers stacked together. We can see the architecture of ELMO in figure 2.9. The ELMo vector assigned to a token or word is actually a function of the entire sentence containing that word. Therefore, the same word can have different word vectors under different contexts. ELMo word vectors take the entire input sentence into account to compute the word embeddings.

Transformers and Self-Attention

The approaches that have been analyzed so far exclude parallelization during training process, leading to threatening and computationally expensive levels reached by memory constraints. Thanks to *attention* ([49]), which computes the global dependencies on its own, recurrence is completely kept out of the process. Given a word at the position i inside an input sentence X of length T, the quantity

$$q_i = \mathcal{Q}\left(x_i, \theta_q\right) \tag{2.15}$$

is computed; therefore, if the whole input sentence is used, both quantities

$$K = \mathcal{K}(X, \theta_k), \quad and \quad V = \mathcal{V}(X, \theta_v) \tag{2.16}$$

get computed for the entirety of words in the sentence. The terms are then q (query), k (key) and v (value). The attention system for a given word representation x_i at position i is:

$$attention(q_i, K, V) = \sum_{j=1}^{T} similarity(q_i, k_j)\, v_j \tag{2.17}$$

In Vaswani et al. ([49]) and in all major techniques adopted recently, the applied mechanism is the scaled dot product attention. The main equation is the following:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{D}}\right) V \tag{2.18}$$

In pictures 2.10 and 2.11, it is possible to see visual representations of the Transformers architecture and mechanism.

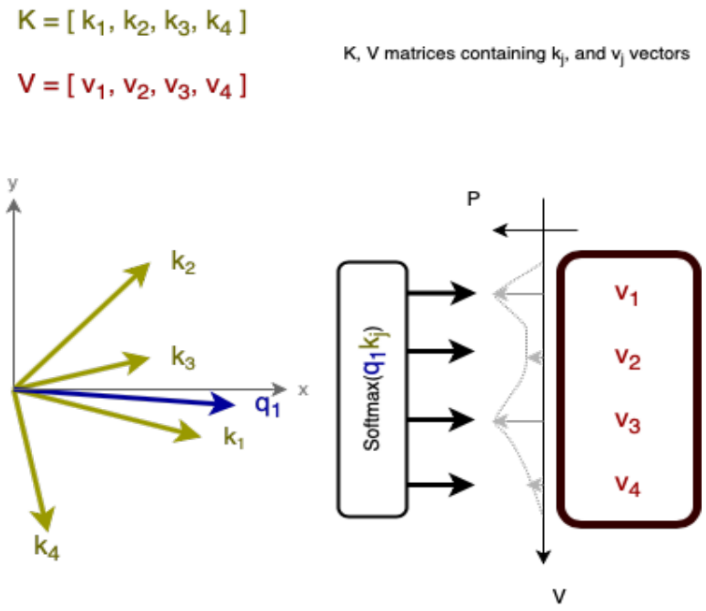

Figure 2.10: Visualization of an example attention mechanism operating on a two-dimensional space, where the first word's query gets compared with all the other terms' keys including itself, resulting in similarity scores.

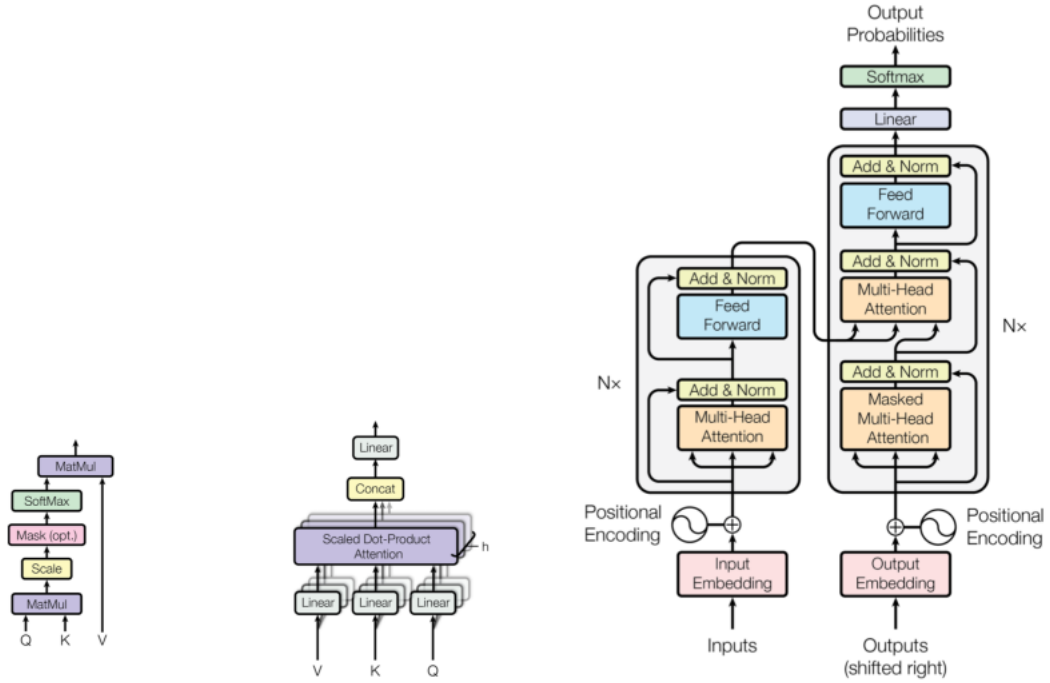

Figure 2.11: Transformers' architecture, with a single attention head alongside a multi-attention block on the left, and the full encoder-decoder system on the right.

BERT

Devlin et al. ([74]) took advantage of the Transformers' innovation brought by Vaswani et al. ([49]) and applied it to language modeling tasks, leading to the birth of **BERT**. BERT (*Bidirectional Encoder Representations from Transformers*, picture 2.12) is a language representation model that obtained state of the art results on several tasks. This model adopts self-attention blocks to record complex and long interdependencies among words and terms in a text: BERT can be pre-trained and successfully used in transfer learning pipeline (like we did in our experiments) to employ and adapt the model on different tasks. Technically speaking, BERT implements a *word piece tokenizer*. This tokenizer, given for example the word *beginning*, returns the tokens:

$$\{beginning\} = \{begin, ning\}. \tag{2.19}$$

One of the most important things here, is that three special tokens ([CLS], [MASK] and [SEP]) are added to the original word:

1. The [CLS] token is the first one in a sequence;
2. The [SEP] token is the one added at the end of the first sentence;
3. the [MASK] token is used to mask (hide) the original token to be predicted.

The representation of a sequence of two sentences will be:

$$\{[\mathrm{CLS}], \mathrm{X}_1, \cdots, \mathrm{X}_n, [\mathrm{SEP}], \mathrm{Y}_1, \cdots, \mathrm{Y}_m\} \tag{2.20}$$

where X_i is the representation of a token contained in the first sentence, and Y_j the representation of a token contained in the second sentence.

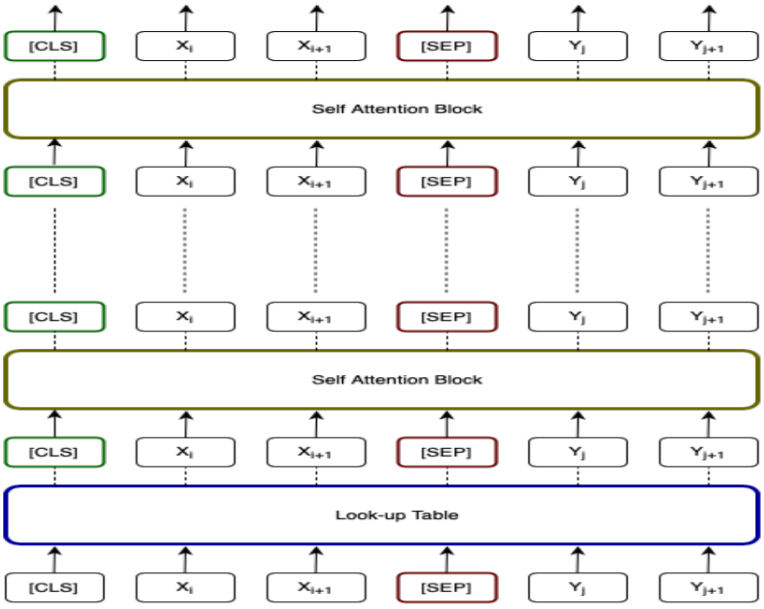

Figure 2.12: Visualization of BERT model, with each layer's representation.

XLM-RoBERTa and Zero Shot Learning

Working with NLP can be difficult when dealing with non-English languages. The advent of transfer-learning and pre-trained models transformed the whole NLP eco-system deeply. The majority of pre-trained models available nowadays, despite providing great results, are mostly limited to English. It is surely possible to create a custom language model from scratch for a specific language, but doing something similar for every language is really complex. The solution to this problem is **zero shot learning**, where the model is first fed with data in a particular language and then the acquired knowledge is transferred to other languages. The most famous and efficient state-of-the-art multilingual model nowadays is **XLM-RoBERTa**, obtained by Liu et al., [75] and Conneau et al ([76]). It was released by the Facebook AI team in November 2019, and it is a Transformer-based language model which relies on the *masked language model* objective to successfully process text from 100 separate languages ([83]). XLMRoBERTA is the most efficient state-of-the-art multilingual model available at the moment.

2.3 Temporal Language Evolution

Human languages change over time, due to a variety of linguistic and non-linguistic factors and at all levels of linguistic analysis ([159]). In the field of theoretical (diachronic) linguistics, much attention has been devoted to expressing regularities of linguistic change. For instance, laws of phonological change have been formulated (e.g., Grimm's law or the great vowel shift) to account for changes in the linguistic sound system. When it comes to lexical semantics, linguists have studied the evolution of word meaning over time, describing so-called lexical semantic shifts or semantic change, which Bloomfield (1933) [102] defines as "innovations which change the lexical meaning rather than the grammatical function of a form." Researchers have realized that "understanding how words change their meaning over time is key to models of language and cultural evolution"

(Hamilton et al., [25]).

2.3.1 Semantic Change Computation

Semantic change computation has long been of interest to both academic circle and the general public. Based on the observations that language is always changing and a dynamic flux ([16]; [17]; [18]), linguists have formulated different theories and models searching for rules and regularities in semantic change, such as the *Diachronic Prototype Semantics* (Geeraerts, [19], [20]), the Invited Inference Theory of Semantic Change (Traugott and Dasher, [21]), and semantic change based on metaphor and metonymy (Heine, Claudi, and Hünnemeyer, [22]). People need to be aware of language change in their daily language use. A large number of websites and books are devoted to etymology studies (Jatowt et al., [23]). Despite the fact that at the moment no published research is available on how the incorporation of semantic change might impact the performance of NLP systems, the potential of semantic change computation to Natural Language Processing (NLP) is huge (Jatowt et al., [23]; Mitra et al., [24]). Semantic change computation deals with the dynamic change of semantics, including the phenomenon of polysemy, a perennial challenge for NLP. A better understanding of polysemy should help improve any NLP task that involves semantic processing. For instance, applications for semantic search can increase the relevance of the query result by taking into account the new senses of words (Mitra et al., [24]; Yao, Sun, Ding, Rao, and Xiong, [26]). Semantic change computation also plays a role in social computing, such as determining the diachronic change of the popularity of brands and persons and the most famous athletes at different times (Yao et al., [26]). Historically, much of the theoretical work on semantic shifts has been devoted to documenting and categorizing various types of semantic shifts (Bréal, 1899 [103]; Stern, 1931 [104]; Bloomfield, 1933 [102]). The categorization found in Bloomfield (1933) [102] is arguably the most used and has inspired a number of more recent studies (Blank and Koch, 1999 [105]; Geeraerts, 1997 [106]; Traugott and Dasher, 2001 [107]). The driving forces of semantic shifts are varied, but include linguistic, psychological, sociocultural or cultural/encyclopedic causes (Blank and Koch, 1999 [105]; Grzega and Schoener, 2007 [108]). Linguistic processes that cause semantic shifts generally involve the interaction between words of the vocabulary and their meanings. This may be illustrated by the process of *ellipsis*, whereby the meaning of one word is transferred to a word with which it frequently co-occurs, or by the need for discrimination of synonyms caused by lexical borrowings from other languages. Semantic shifts may be also be caused by changes in the attitudes of speakers or in the general environment of the speakers (Bloomfield, [102]). Thus, semantic shifts are naturally separated into two important classes: linguistic drifts (slow and steady changes in core meaning of words) and cultural shifts (changes in associations of a given word determined by cultural influences). Gulordava and Baroni (2011) [114], for instance, showed that distributional models capture cultural shifts, like the word *'sleep'* acquiring more negative connotations related to sleep disorders domain, when comparing its 1960s contexts to its 1990s contexts. Researchers studying semantic shifts from a computational point of view have empirically shown the existence of this division (Hamilton et al., 2016b - [25]). In the traditional classification of Stern (1931) [104], the semantic shift category of substitution describes a change that has a non-linguistic cause, namely that of technological progress. This may be exemplified by the word *car* which shifted its meaning from non-motorized vehicles after the introduction of the automobile. Diachronic corpora provide empirical resources for semantic change

computation. The construction of diachronic corpora is based on different factors such as size, balance, and representativeness of the corpora in question. For semantic change computation, only those corpora that are designed to be informative of semantic change should be used, and only those semantic aspects that are contrastive in the corpora should be studied (Sinclair, ([27]). Figure 2.13 lists the most famous corpora used for semantic change computation researches in literature. The most frequently used corpus is the Google Books Ngram Corpus (Y. Lin, Michel, Aiden, Orwant, Brockman, and Petrov, [28]), due to its size, temporal range, types of language included in the corpus and public availability. For our work, the most important corpora are without any doubt the following ones:

1. Corpus of Historical American English (CoHa);
2. British National Corpus;
3. New York Times Corpus

Names	Language and Times	Used in
Google Books Ngram Corpus	English, French, Spanish, German, Chinese, Russian and Hebrew (From 1500s to 2008)	Michel et al. (2011), Gulordava and Baroni (2011), Hamilton et al. (2016), Wijaya and Yeniterzi (2011), Jatowt et al. (2014), Yang and Kemp (2015), Dubossarsky et al. (2017)
Corpus of Historical American English (COHA)	American English (1810s-2000s)	Neuman, Hames, and Cohen (2017), Hamilton et al. (2016)
Google Books Syntactic Ngram Corpus	English (1520-2008)	Mitra et al. (2015) Goldberg and Orwant (2013)
Helsinki Corpus of English Texts	Old English (till 1150) Middle English (1150-1500) Early Modern English (1500-1710)	Sagi et al. (2009), Sagi, Kaufmann, and Clark (2011),
TIME corpus from BYU	American English (1923-2006)	Martin et al. (2009)
New York Times Corpus	American English (1987-2007)	Rohrdantz et al. (2011)
Corpus from New York Times	American English (1990-2016)	Yao et al. (2017)
DATE corpus[1]	English (1700-2100)	Frermann et al. (2016)
Newspaper Corpus from People's Daily	Modern Chinese (1946-2004)	Tang et al. (2016)
Newspaper Corpus from Le Monde	Modern French (1997-2007)	Boussidan and Ploux (2011)
Parole Corpus for modern Swedish and Swedish Literature Bank for Swedish in the 19th century		Cavallin (2012)
British National Corpus for late 20th century and ukWaC for 2007		Lau et al. (2012)
Twitter Corpus, 1% of the Twitter data from 2012 to 2013		Mitra et al. (2015)

Figure 2.13: Main diachronic corpora in literature.

The availability of large corpora have enabled the development of new methodologies for the study of lexical semantic shifts within general linguistics (Traugott, 2017 [109]). A key assumption is that changes in a word's collocational patterns reflect changes in word

meaning (Hilpert, 2008 [110]), thus providing a usage-based account of semantics (Gries, 1999 [111]). The usage-based view of lexical semantics aligns well with the assumptions underlying the distributional semantic approach (Firth, 1957 - [113]) often employed in NLP. To summarize, semantic shifts are often reflected in large corpora through change in the context of the word which is affected by the shift, as measured by co-occurring words.

2.3.2 Vectors' Comparison Across Time and Procrustes

Separate word embedding models can be trained using time-specific corpora containing texts from several different time periods. These models are thus time-specific. However, comparing word vectors across different models is not an easy task. It usually does not make sense to, for example, directly calculate cosine similarities between embeddings of the same word in two different models ([159]). Therefore, even when trained on the same data, separate learning runs will produce entirely different numerical vectors. This is even more true for models trained on different corpora: even if words' meaning is totally stable, the direct cosine similarity between its vectors from different time periods can still be quite low, simply because the random initializations of the two models were originally different. To solve this problem, Kulkarni et al. (2015) [115] suggested that before calculating similarities, the first thing to do should be aligning the models to fit them in one vector space, using linear transformations and preserving general vector space structure. After that, cosine similarities across models become meaningful and can be used to successfully identify semantic shifts. They also proposed the construction of word embeddings' time series over time (Taylor, 2000 [116]). The idea of aligning diachronic word embedding models using a distance-preserving projection technique was also proposed by Zhang et al. (2015) [117]. Later, always Zhang et al. (2016) [118] expanded on this by adding the so called 'local anchors': they used both linear projections for the whole models and small sets of nearest neighbors in order to map the query words to their correct temporal counterparts. Instead of aligning their diachronic models using linear transformations, Eger and Mehler (2016) -[119] compared word meaning using so-called 'second-order embeddings': these are the vectors of words' similarities to all other words in the shared vocabulary of all models. This technique does not require any transformations, since it is basically based on a word's position compared to other words. Hamilton et al. (2016c) [58] and Hamilton et al. (2016b) [25] showed that orthogonal procrustes approaches can be used simultaneously: both 'second order embeddings' and orthogonal Procrustes transformations were applied to align diachronic models. Procrustes alignment ([120]) takes on the task of aligning two sets of points in high dimension (which has many applications in natural language processing), through the joint estimation of an orthogonal matrix and a permutation matrix. A stochastic algorithm is proposed to minimize the cost function on large scale problems, and the method is finally evaluated on the problem of unsupervised word translation. This is done by aligning word embeddings trained on monolingual data. Procrustes analysis learns a linear transformation between two sets of matched points $X \ R\ n{\times}d$ and $Y \ R\ n{\times}d$. If the correspondences between the two sets are known, then the linear transformation can be recovered by solving the following least square problem:

$$\min_{\mathbf{W} \in R^{d \times d}} \|\mathbf{XW} - \mathbf{Y}\|_2^2 \tag{2.21}$$

This technique has been successfully adopted in many different fields, such as learning a linear mapping between word vectors in two different languages with the help of a bilingual lexicon (Mikolov et al., 2013) [122]. Even Xing et al. (2015) [123] have shown that orthogonal transformations are well suited to the mapping of word vectors. The corresponding orthogonal Procrustes corresponds to the following optimization problem:

$$\min_{\mathbf{Q} \in \mathcal{O}_d} \|\mathbf{XQ} - \mathbf{Y}\|_2^2 \tag{2.22}$$

where O_d is the set of orthogonal matrices. The distances between points are ensured to remain unchanged by the transformation, thanks to the orthogonality constraint.

2.3.3 CADE - Compass Alignment

CADE (*Compass aligned distributional embeddings*), is an extension of CBOW (Continuous Bag of Words) Word2vec model that can be applied to sliced corpora in order to create a set of aligned word embeddings ([65]). CADE can be also applied on top of the Skip-gram Word2vec model. It was empirically found and proved that CBOW is able to produces models that present better performance than Skip-gram, when doing experimental evaluations on small datasets. Therefore, CADE has been mainly used on top of CBOW. The training process of CADE is divided into two phases, which are represented in shown 2.14:

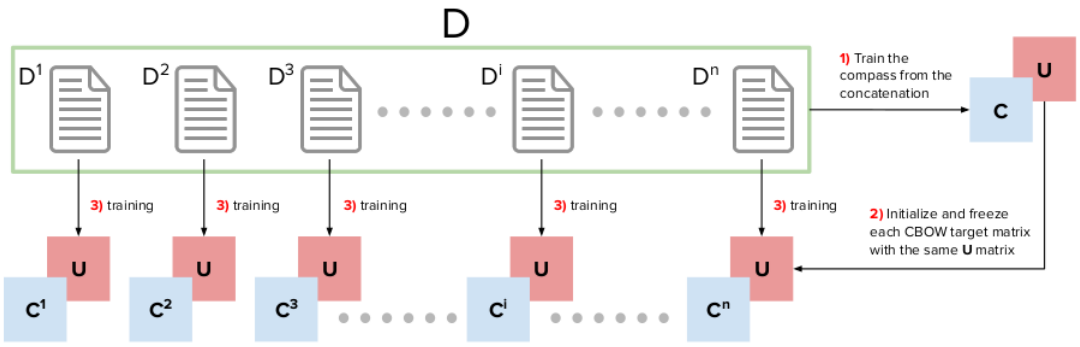

Figure 2.14: Visual representation of CADE model.

In the first step, two compass matrices C and U are constructed by applying the original CBOW model on the concatenation of the document in the collection. D; C and U represent the set of compass context embeddings and compass target embeddings. Then, for each specific slice D_i, the context embedding matrix C_i is created by initializing the output weight matrix of the neural network with the target embeddings from the matrix U that were trained before. The CBOW algorithm is run using the specific slice D_i, and here the target embeddings of the output matrix U don't get modified since the layer is frozen. The context embeddings, instead, are updated in the input matrix C_i. This process is adopted through all the slices D_i; each C_i matrix represents the word embeddings for the slice i. The two main processes that constitute the CADE model are the update of the matrix for each slice and the interpretation of the update function.

Given a slice D_t, the second step of the training can be formalized for a single training sample $\langle w_k, \gamma(w_k)\rangle \in D^i$ as this specific optimization problem:

$$\max_{\mathbf{C}^i} \log \mathrm{P}(w_k \mid \gamma(w_k)) = \sigma\left(\mathbf{u}_k \cdot \mathbf{c}^i_{\gamma(w_k)}\right) \tag{2.23}$$

where $\gamma(w_k) = (w_{j_1}, \cdots, w_{j_M})$ stands for the M words in the context of wk which appear in D_i, $uk \in U$ is the target embedding of the word w_k (the fixed one and not updated one), and

$$\mathbf{c}^i_{\gamma(w_k)} = \frac{1}{M}\left(c^i_{j1} + \cdots + c^i_{j_M}\right)^\top \tag{2.24}$$

is the mean of the context embeddings $c_{j_{im}}$ of the contextual words w_{j_m}. Negative sampling is used to compute the softmax function *sigma*. The biggest difference from the original CBOW model, is that C_i is the only weight matrix that gets optimized in this phase (while U is the compass embedding which isn't updated during training). The probability that given the context of a word w_k in a specific slice i, that word can be predicted using the target matrix U, is maximized during the training. At the end of process, the embeddings of the finally trained context matrix are extracted and utilized. They are at this point already aligned, thanks to the shared target embeddings used as a compass during the independent training.

2.4 Computational Approaches to Metaphor Detection

2.4.1 Methods Background

Metaphors are pervasive in language use, and their detection and interpretation are crucial to language processing (Group, 2007 [10]; Turney et al., 2010 [11]; Shutova, 2015 [7]). Most of the computational work on metaphors has focused on their identification and interpretation through several techniques and models, such as clustering (Shutova and Sun, [13]), LDA topic modeling (Heintz et al., [14]), tree kernels (Hovy et al., [15]), but all from a purely synchronic perspective. Ekaterina Shutova ([7]) studied several approaches to metaphor detection: these are regression, random forest and support vector machine (SVM) models, neural networks and word embeddings, and clustering approaches. The latter started out as supervised and ended up as unsupervised approaches to metaphor detection. Figure 2.15 shows a timeline of a few main clustering approaches in literature.

The two main metaphor detection approaches used in literature are:

1. **Metaphor Detection in Single Sentence**: local contextual clues within a short text are used in order to detect metaphors. Turney et al. (*Literal and metaphorical sense identificaion through concrete and abstract context*, [29]) used logistic regression and abstractness of context, considering metaphors as a method of transferring knowledge from a familiar domain to an unfamiliar one. Tsvetkov et al. (*Metaphor detection with cross-lingual model transfer*, [30]), instead, used a random forest classifier with conceptual semantic features such as abstractness, imageability and semantic supersenses. His main hypothesis was that metaphors are conceptual figures rather lexical ones.

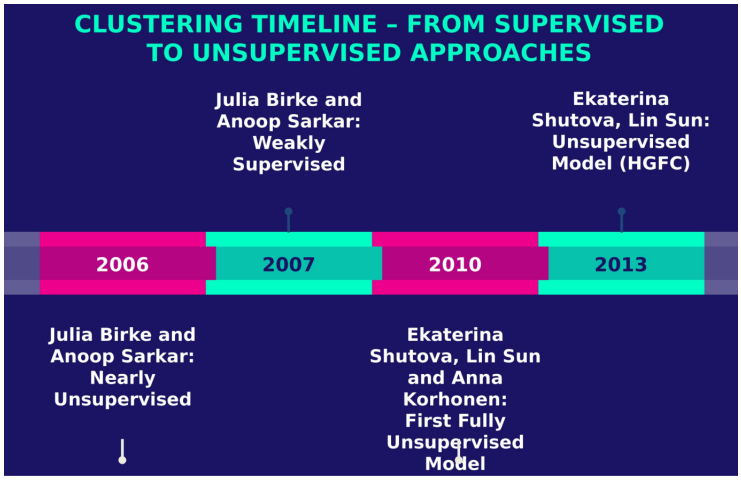

Figure 2.15: Main clustering approaches timeline

2. **Metaphor Detection in Discourse**: the global context of a discourse is used in order to detect metaphors, thanks to a combination of global and local features. Klebanov et al. (*Different texts, same metaphors: unigrams and beyond* [31]) classified each content-word token in a text as a metaphor or non-metaphor, using logistic regression models and through features such as unigrams (the most effective ones), part-of-speech, concreteness and topic models. Jang et al. (*Metaphor detection with topic transition, emotion and cognition in context*, [32]) relied instead on a SVM model, where the features were the topic transition patterns between sentences containing metaphors and their contexts.

Some models interpret the identified metaphors, paraphrasing them into their literal counterparts, so that they can be better translated by machines. This allows *metaphor identification and interpretation* in whole sentences, as shown by Rui Mao et al., in *Word Embedding and WordNet Based Metaphor Identification and Interpretation*, [33].

1. **Metaphor Identification** - this approach is often based on: violation of selectional preferences (Fass, [34]); clustering (Birke and Sarkar, [35] and Shutova et al., [36]); lexical relations in WordNet (Krishnakumaran and Zhu, [37]); contrast between literal and non-literal use of a target expression in text and source-target domain mappings (Mason, [38]).

2. **Metaphor Interpretation** - Ekaterina Shutova, in 2013, performed Metaphor Identification and Interpretation simultaneously. At first, much of the computational work had focused on detecting and uncovering the intended meaning behind metaphors: Klebanov & Flor ([39]) paid attention to the motivations behind metaphor use, and in particular to the moderate-to-strong correlation between percentage of metaphorically used words in an essay and writing quality score.

2.4.2 End-to-End Sequential Metaphor Identification

In *Neural Metaphor Detection in Context*, Ge Gao et al. ([41]) presented end-to-end neural models for detecting metaphorical word used in a context. They showed that BiLSTM models which operate on complete sentences work well for the task of metaphor detection.

These models establish a new state-of-the-art on existing verb metaphor detection benchmarks, and show strong performance on simultaneously predicting the metaphoricity of all words in a running text. Two common task formulations are investigated:

1. Given a target verb in a sentence, classifying whether it is metaphorical or not;
2. Given a sentence, detecting all of the metaphorical words (independent of their POS tags).

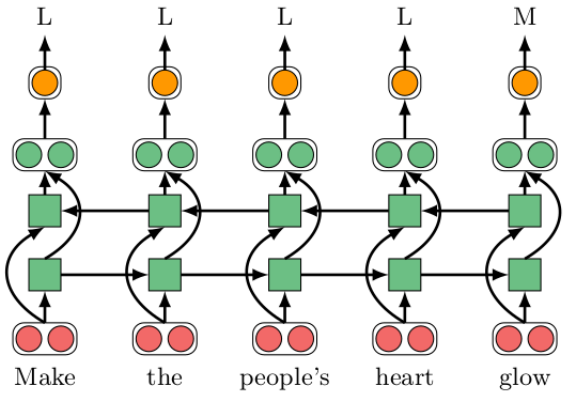

Figure 2.16: A sequence labeling model for metaphor detection: each word in a sentence is classified.

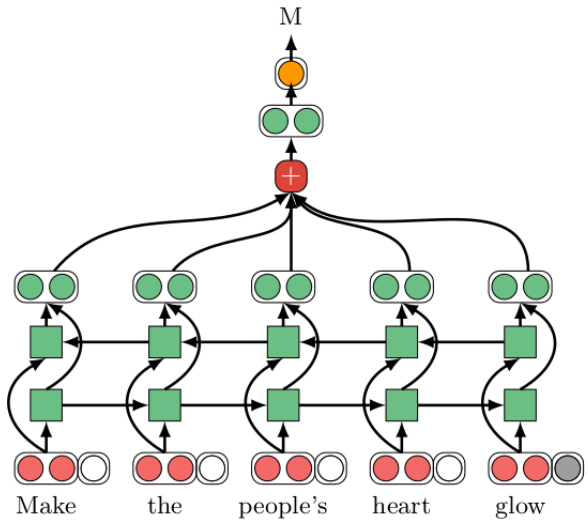

Figure 2.17: A classification model for metaphor detection: in this case, only a single word per sentence is labeled as metaphorical or not.

Relatively standard architectures based on bi-directional LSTMs (Hochreiter and Schmidhuber, [42]) augmented with contextualized word embeddings (Peters et al., [43])

perform surprisingly well on both tasks, even with small amount of training data. In order to conduct our experiments with metaphors and temporal word embeddings, two models proposed by Ge Gao et al. [41] have been modified and used:

1. **Recurrent Neural Network Hidden GloVe (RNN HG)**: Based on the MIP (the metaphor is classified by the contrast between a word's contextual and literal meanings) *metaphor identification procedure* ([10]), here the GloVe embedding serves as the literal representation and it joins the hidden state from the BiLSTM (contextual representation);

2. **Recurrent Neural Network Multi-Head Context Attention (RNN MHCA)**: based on Wilks' SPV procedure [137] (the intuition behind *selectional preference violation* is that metaphoricity is successfully identified by detecting the incongruity between a target word and its context), here a label prediction is conditioned on a hidden state of a target word and its attentive context representation.

In MIP ([10]) and SPV ([137]) procedures, the modelling of the contrast between literal and contextual representations (meanings) of metaphors is theroretically similar: humans are usually able to infer the contextual meanings of a word conditioned on its context. In BiLSTM, the hidden states, used as contextual meaning representations, are extracted through the forward and the backward contexts and itself: Graves and Schmidhuber's work ([44]) was the first one ever with bidirectional training being applied on a Long Short Term Memory (LSTM) network. The input is presented forwards and backwards to two separate recurrent networks, both of which are connected to the same output layer. This is even better than introducing a delay between inputs and their associated targets giving the network a few time steps of future context, a trick that is usually applied to overcome the most typical recurrent neural networks' limit: analyzing data only in one single direction (the past). As anticipated earlier, pretrained GloVe representations of each dataset are used as the input, aka literal meaning representation, as words have been embedded with their most common senses (trained on Web crawled data).

Recurrent Neural Network Hidden GloVe

In bidirectional networks models ([46]; [47]), the input is presented forwards and backwards to two separate recurrent nets, both of which are connected to the same output layer. Bidirectional recurrent neural networks (BRNNs) have lead to hugely improved scores in sequence learning tasks and procedures. In the RNN HG model, GloVe and ELMo (the latter is applied in the same way as in [41] and concatenated with GloVe [43]) input representations feed the BiLSTM network. ELMO uses character convolutions to create the initial representations in the "look-up" table and computes different representations at each layer. We call the first layer "look-up" table, as it contains representations for every word, or subword unit, in the model's vocabulary. Every other layer consists of two LSTM networks serving as a Forward language model, and a Backward language model used in conjunction.

These language models take as input the representation of each word and generate hidden states that serve as word representations, Then literal and contextual representations get compared in the so called *comparison stage*. The BiLSTM has a hidden state

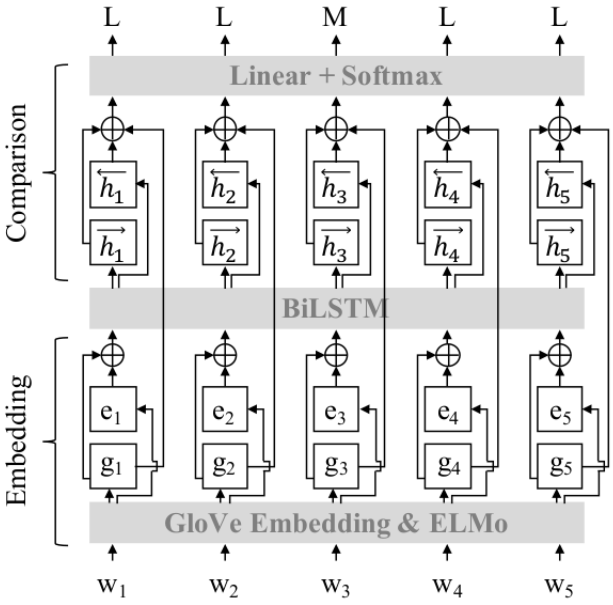

Figure 2.18: RNN_HG model architecture based on MIP procedure, with tensors concatenated along the last dimension.

which is, as stated before, the contextual representation of the sentences: this hidden state h_t joins the input literal representations and is expressed by the equation:

$$h_t = f_{BiLSTM}([g_t; e_t]\, \vec{h}_{t-1}, \vec{h}_{t+1}\Big) \tag{2.25}$$

The two concatenated vectors are two different representations of the same word (literal and contextual), and they are located in two different encoding spaces. The last step consists in the softmax function σ, which calculates the probability of a label prediction $\hat{y}$ for a target word at position t, conditioned on both its contextual and literal meaning representations. This is expressed by the equation:

$$p\left(\hat{y}_t \mid h_t, g_t\right) = \sigma\left(w^\top \left(h_t; g_t\right) + b\right) \tag{2.26}$$

Recurrent Neural Network Multi-Head Context Attention

Attention was first introduced by Bahdanau et al. ([48]), alongside a recurrence based encoder-decoder machine translation network. One of these networks acted as an encoder for the input and the other one as a decoder, taking as input the encoding, and the previous output. Given a target translation Y of length T, the objective is expressed by the equation:

$$\mathcal{P}(Y) = \prod_{t=1}^{T} \mathcal{P}\left(y_t \mid y_1, \cdots, y_{t-1}; c\right) \tag{2.27}$$

where the probability of a word at time-step t is computed as follows:

$$\begin{gathered} \mathcal{P}\left(y_t \mid y_1, \cdots, y_{t-1}; c\right) = \mathcal{G}\left(y_{i-1}, s_i, c\right) \\ c = f\left(h_1, \cdots, h_N\right), and h_i = g\left(x_i, h_{i-1}\right) \end{gathered} \tag{2.28}$$

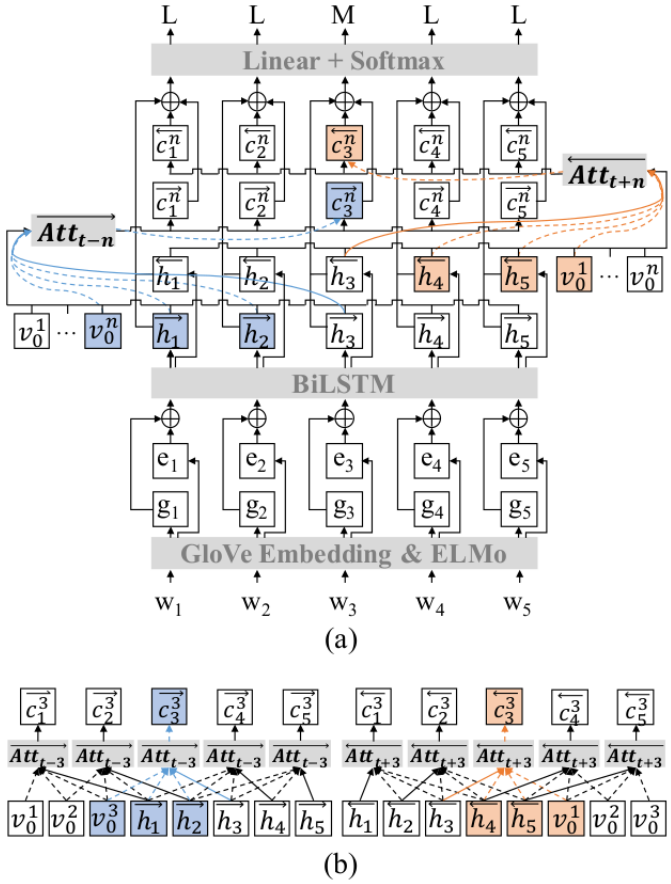

Figure 2.19: RNN_MHCA model architecture based on SPV procedure, with attention mechanisms on a window of n context words.

In our specific case, the target word representation is compared with its context by concatenating them, just like GloVe and ELMo were in the Hidden Glove model. The BiLSTM hidden state is formed by the target word representation h_t, while context has left and right side attentive context representations. The context is derived through the *multi-head contextual attention procedure* ([139]). Word and context representations of the same word are concatenated: if the target word representation is located in the same encoding space as the one of the attentive context representation, final results will be better. Multi-head self-attention ([49]) encodes a target word by its context, while MHCA model calculates the context representation by attending to a target word. In RNN MHCA model, the probability of a label prediction is expressed by the equation:

$$p\left(\hat{y}_t \mid h_t, c_t^n\right) = \sigma\left(w^\top \left[h_t; c_t^n\right] + b\right) \tag{2.29}$$

where a label prediction is conditioned on a hidden state of a target word and its attentive context representation.

Comparison Between RNN-Based Models

The objective functions of the RNN HG and RNN MHCA models (2.26, 2.27) are different: this leads to different errors being backpropagated to the BiLSTM during the training phase, and different hidden states (even if for the same word of a sentence) being

received as well. Let's understand this better by briefly comparing the two models using an example sentence: *car drinks gasoline*.

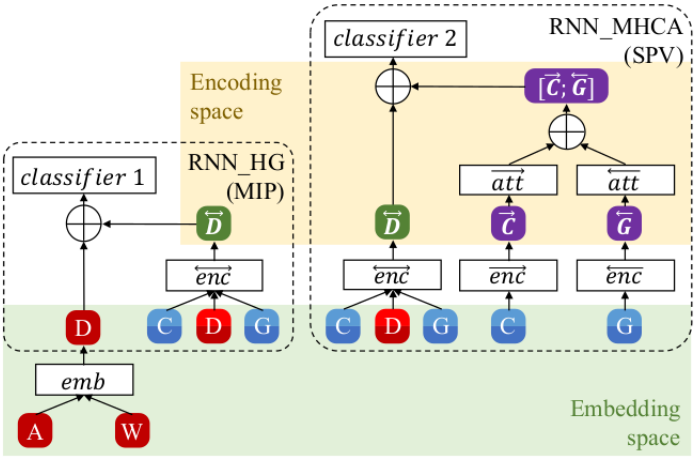

Figure 2.20: A comparison between RNN HG and RNN MHCA. C is car. D is drinks. G is gasoline. A is animal. W is water. emb is GloVe embedding. enc is BiLSTM encoding. att is an attention mechanism. In embedding space, the lighter part of a node is ELMo embedding, while the darker part is GloVe embedding.

The vectors assigned to the three words are distant from each other because they come from non-literally related domains. If we start analyzing the RNN HG model and we take for example the word *drinks*, it is characterized by two different vectors/embeddings:

1. A **contextual** one, which is determined by the other words in the sentence and captures the meaning of the word (in this case, *drinks*) in context via BiLSTM encoding;

2. A **literal** one.

These two embeddings are concatenated as explained before. The RNN HG classifier, after training, understands whether these two aforementioned vectors portray similar or different meanings: in the first scenario the sense will be literal, while in the latter a metaphor will be detected. In the RNN MHCA model's architecture instead, the contextual embedding is used as the target word representation, and gets concatenated with the word's attentive context representation. These two representation can be located in the same encoding space (leading to better results) or in two different ones, as it happens for the RNN HG model (since in that model we have two representations of the same word and not the word and its context). As far as general tasks are concerned, the RNN MHCA gives slightly better results because it models the contrast between the metaphor and its context in a single-metaphor sentence.

Chapter 3

RNN-based Models with Temporal and Other Embeddings

3.1 Temporal Metaphor Detection Experiments

The way metaphors develop across time, and whether the shift of a word's literal meaning to a figurative one can be automatically detected and modelled is a relatively little investigated aspect. One tricky characteristic of metaphors consists in their dynamic nature: new metaphors are created all the time. In Del Tredici, Nissim and Zaninello's work [12], consecutive vector spaces are built from a diachronic corpus of Italian language and used to compare a term's cosine similarity to itself in different time spans. The following are some of the most relevant notions that we learn through Del Tredici et al.'s ([12]) work:

1. If a metaphorical meaning is acquired by a term at a certain point in time, the context of use of that term will, at least partially, change;

2. (Dis)similarity of contexts is measured relying on the distributional semantics approach ([133]) and on the term's vector representations, and derived from the Zanichelli dictionary;

3. The meaning of a word is represented by vectors that encode the contextual information of that word in a corpus (Turney et al., 2010) [11] – all vectors representing words are included in a distributional semantic space in which similar words are represented by vectors that are close in that space, while different ones are distant;

4. Linguistic drift is defined as organically recurring shifts in the meaning of a given word as a result of regular processes of semantic change.

To this regard, an interesting example about the Italian term "talebano" ('Taliban') is made by Del Tredici et al. ([12]): the word, which was previously only used to refer to the Islamic fundamentalist political movement founded in the Nineties in Afghanistan, has come to more generally define someone who is extreme in his or her positions, for example regarding food, use of medicines, ecc. If the metaphorical meaning becomes commonly used with time, it might get recorded in reference dictionaries. In the first part of this Chapter, we used the previously introduced RNN HG and RNN MHCA models with different datasets and word embeddings (GloVe, ELMO, Wikipedia), and especially with temporal word representations such as HistWords - SGNS and CADE-aligned models, to understand whether the latter can enhance metaphor detection task's performances.

The two aforementioned models are used to perform metaphor detection as a *sequence classification* task, in order to detect single metaphorical words (nouns, verbs, adjectives and so on) inside whole sentences. In the first part of this Chapter, we will try to answer the following research questions:

1. Can we improve the state of the art results for metaphor detection obtained with standard GloVe representations (combined with ELMO vectors) by using other word embeddings, especially temporal ones such as HistWords - SGNS?

2. Are there any observable patterns that can lead us to presume that metaphor detection tasks performed with temporal embeddings and representations impacts datasets with known temporal connotations more than others?

3. Are representations obtained through CADE and Compass alignment more effective for metaphor detection than embeddings aligned with traditional methods (e.g.: Procrustes)?

4. Are specific word embeddings' architectures more effective for metaphor detection tasks than others?

3.2 Datasets

The experiments and approaches have been performed on several metaphor datasets, including three of the most used ones in literature for metaphor detection. The table in figure 3.1 provides a glimpse of their main characteristics; in the following sub-sections, we will analyze them in detail.

Dataset	N. of sentences	Train/test/split	How it was built	Clear temporal connotation
MOH-X	646	Cross-validation required	Derived from the subset of MOH dataset used by Ekaterina et. al. The verbs are annotated for metaphoricity and they come from WordNet.	No
VUAsequence	6323	Yes	117 fragments sampled across 4 genres from the British National Corpus (acdemic, news, conversation and fiction).	Yes 1985-1994
TroFi	3737	Cross-validation required	The sentences (each one with a single annotated target verb) are taken from the '87-'89 Wall Street Journal corpus (WSJ).	Yes 1987-1989

Figure 3.1: Datasets' overview table.

3.2.1 MOH-X

MOH-X dataset ([50]) is derived from the subset of MOH dataset used by Ekaterina et. al ([51]). Mohammad et al. annotated different senses of WordNet ([52]) verbs for metaphoricity. They extracted verbs that had between three and ten senses in WordNet and the sentences exemplifying them in the corresponding glosses.

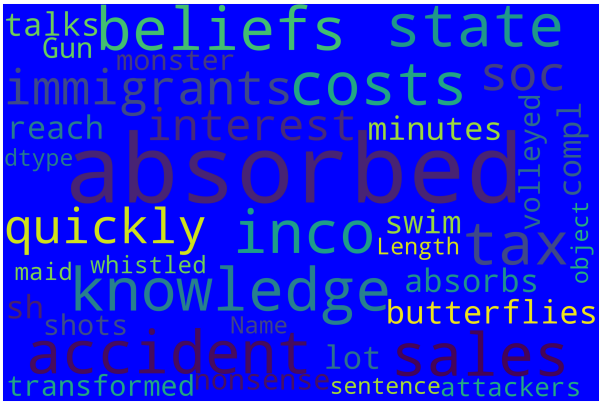

Figure 3.2: A wordcloud visualization for MOH-X corpus.

In WordNet, each verb sense corresponds to a synset, which consists of a set of near-synonyms, a gloss (a brief definition), and one or more example sentences that show the usage of one or more of the near-synonyms. These sentences are referred to as the verb-sense sentence, or just sentences ([50]). The portion of the sentence excluding the target verb is called the context. Each pair of target verb and verb-sense sentence is referred to as an instance. The following is an example of instance extracted from WordNet (quote 3.2.1):

> *" The Turks erased the Armenians in 1915 ."*
>
> Sentence taken from MOH-X dataset Target verb: *erase*

Here, *erase* was used metaphorically. We will refer to such instances as metaphorical instances. Another instance of the verb erase, corresponding to a different sense, is shown below (quote 3.2.1):

> *" Please erase the formula on the blackboard – it is wrong !"*
>
> Sentence taken from MOH-X dataset Target verb: *erase*

The aforementioned instance contained a literal use of *erase*. The 1639 verbs initially used in the sentences were annotated for metaphoricity by ten annotators via the crowd-sourcing platform CrowdFlower; Mohammad et al. selected the verbs that were tagged by at least 70 per cent of the annotators as metaphorical or literal to create the dataset. The final result was a dataset of 647 verb-noun pairs, 316 metaphorical and 331 literal. Examples that have no mapping in the original MOH dataset were discarded; train-validation-test split is not available. The majority of non-target words are literal.

3.2.2 VUA

The dataset consists of 117 fragments sampled across four genres from the British National Corpus (Academic, News, Conversation and Fiction). ([53]).

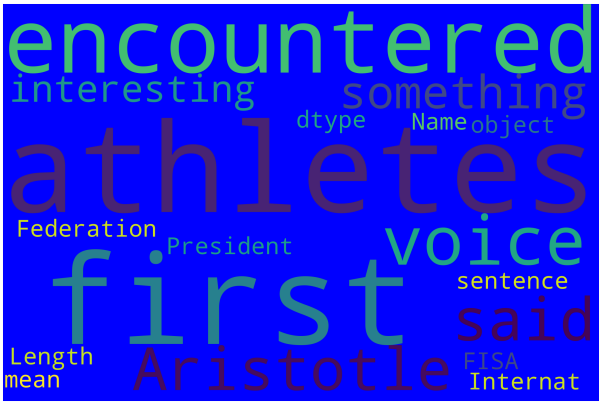

Figure 3.3: A wordcloud visualization for VUAsequence corpus.

Each genre is represented by approximately the same number of tokens, although the number of texts differs greatly. 23% of the text from each genre was randomly sampled to set aside for testing, while retaining the rest for training. The data was annotated using the MIP-VU procedure ([54]): it is based on the MIP procedure (Group, 2007), extending it to handle metaphoricity through reference (such as marking *did* as a metaphor in 'As the weather broke up, so did their friendship') and allow for explicit coding of difficult cases where a group of annotators could not arrive at a consensus. The tagset is rich and organized hierarchically, detecting various types of metaphors, words that flag the presense of metaphors, etc. The vast majority of dataset's sentences come from the decade 1985-1994, while some samples from a decade before (1975).

3.2.3 TroFi

Figure 3.4: A wordcloud visualization for TroFi corpus.

The TroFi dataset is formed by literal and non literal sets, that contain feature lists consisting of the stemmed nouns and verbs in a sentence, with target or seed words and frequent words removed. It is named after TroFi (Trope Finder), a nearly unsupervised clustering method for separating literal and non-literal usages of verbs ([35]). For example, given the target verb *pour*, we would expect TroFi to cluster the sentence *Custom demands that cognac be poured from a freshly opened bottle* as literal, and the sentence

Salsa and rap music pour out of the windows as nonliteral, which, indeed, it does. The TroFi algorithm requires a target set (called original set in Karov Edelman, [55]) – the set of sentences containing the verbs to be classified into literal or metaphorical – and the seed sets: the literal feedback set and the nonliteral feedback set. The frequent word list consists of the 332 most frequent words in the British National Corpus plus contractions, single letters, and numbers ranging from 0 to 10. The target set is built using the '88-'89 Wall Street Journal Corpus (WSJ) [125] tagged using the Ratnaparkhi tagger ([56]) and the Bangalore Joshi SuperTagger ([57]); the feedback sets are built using WSJ sentences containing seed words extracted from WordNet and the databases of known metaphors, idioms, and expressions (DoKMIE), namely Wayne Magnuson English Idioms Sayings Slang and George Lakoff's Conceptual Metaphor List ([126]), as well as example sentences from these sources. TroFi has a lot of non-target words that are metaphorical (but not labeled), and long sentences.

3.3 Embeddings

3.3.1 CoHa - Corpus of Historical American English

The Corpus of Historical American English (CoHa) [134] is the largest structured corpus of historical English, and it is related to many other corpora of English. The CoHa corpus contains more than 400 million words of text from the 1820s-2000s (which makes it 50-100 times as large as other comparable historical corpora of English). The creation of the corpus results from a grant from the National Endowment for the Humanities (NEH) from 2008-2010. We bought the full CoHa corpus for our quantitative approaches, in all its three main formats:

1. **Database**: This is the most robust format, and requires knowledge of SQL ([135]). which allows to perform joins across corpus, lexicon, and sources tables;

2. **Word/lemma/PoS**: Word, lemma, and part of speech are in vertical format, and they can be imported into a database. In most of the corpora, texts are separated by a line with and the *textID*. In COHA, each text is its own file;

3. **Linear text**: This format gives a *textID* for each text, and then the entire text on the same line. Here words are not annotated for part of speech or lemma. Contractions like *can't* are separated into two parts, *ca n't* and punctuation is separated from words.

The third format is the one that we used for our experiments. A very interesting advantage that we have with COHA, is to be able to use built-in synonyms to search for the frequency of the concept of 'beautiful', the adjective 'clever' or the noun 'clean' by decade. To sum up, COHA allows to look at many important and interesting changes:

1. **lexis**, through comparison between historical periods;

2. **syntax** through the part of speech corpus;

3. **semantics (word meaning)**, through synonyms and customized lists.

3.3.2 HistWords - Word Embeddings for Historical Text

HistWords is a collection of tools and datasets for analyzing language change using word embeddings ([136]). The historical word vectors in HistWords were used to study the semantic evolution of more than 30,000 words across four languages, leading to the formulation of the *law of conformity* (words that are used with more frequence change less and are characterized by more stable meanings over time) and *law of innovation* (polysemous words, that have more than one meaning and change at faster rates).[58]

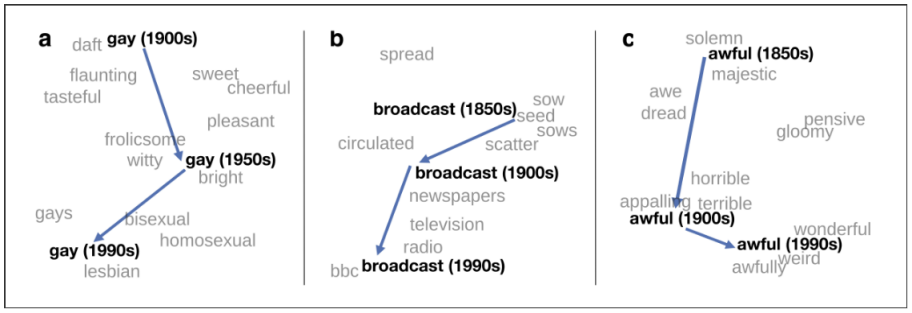

Figure 3.5: Visualization of changes in words' meaning over time.

We used pretrained historical word embeddings for English language spanning all decades from 1800 to 2000. The four main SGNS HistWords embeddings are the following:

1. **All English** (1800s-1990s - from Google N-Grams [64] eng-all): These datasets were created in July 2012 (Version 2) and July 2009 (Version 1) - Each of the files constituting the corpora is compressed tab-separated data. In Version 2 each line consists of *ngram, year, match_ count, volume_ count*;

2. **English Fiction** (1800s-1990s - from Google N-Grams eng-fiction-all): They present the same structure of All English;

3. **Genre-Balanced American English** (1830s-2000s - from Corpus of Historical American English [59]);

4. **Genre-Balanced American English, word lemmas** (1830s-2000s - from Corpus of Historical American English [59]).

These diachronic word embeddings were obtained by aligning over time embeddings originally constructed in each time-period through three main methods (PPMI; SVD; SGNS - word2vec) that represent each word by a vector that captures information about its co-occurrence statistics.([60]). These methods take advantage of the 'distributional hypothesis': word semantics are implicit in co-occurrence relationships ([61]; [62]). Besides, the semantic similarity and distance between two words is approximated by the cosine similarity and distance between their respective vectors ([63]). Models were trained on 6 datasets taken from Google N-Grams ([64]) and the COHA corpus (Mark Davies, [59]). Finally, the diachronic embeddings were aligned using the orthogonal Procrustes method. The Google N-Gram datasets are extremely large. The CoHa corpus contains pre-extracted word lemmas and was originally selected to be genre-balanced and representative of American English over the last two centuries, but it ended up being smaller. Both datasets were aggregated to the granularity of decades.

3.3.3 Contextual Representations for Downstream Tasks

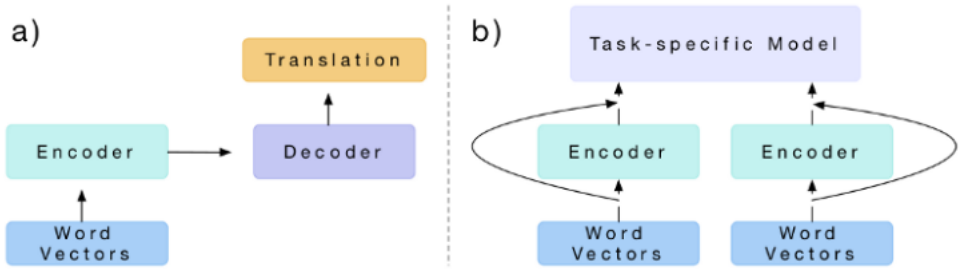

Figure 3.6: a) Training an encoder b) Reusing it for a downstream task. Source: "Learned in Translation: Contextualized Word Vectors" by McCann et al. 2017.

Contextual embeddings, such as ELMo and BERT, move beyond global word representations like Word2Vec and achieve ground-breaking performance on a wide range of natural language processing tasks. Contextual embeddings assign each word a representation based on its context, thereby capturing uses of words across varied contexts and encoding knowledge that transfers across languages. ([160]). In large part, pre-training contextual embeddings can be divided into either unsupervised methods (e.g. language modelling and its variants) or supervised methods (e.g. machine translation and natural language inference). ([160]). The prototypical way to learn distributed token embeddings is via language modelling. A language model is a probability distribution over a sequence of tokens. Given a sequence of N tokens, $(t_1, t_2, ..., t_n)$, a language model factorizes the probability of the sequence as:

$$p\left(t_1, t_2, \ldots, t_N\right) = \prod_{i=1}^{N} p\left(t_i \mid t_1, t_2, \ldots, t_{i-1}\right) \tag{3.1}$$

Language modelling uses maximum likelihood estimation (MLE), often penalized with regularization terms, to estimate model parameters. A left-to-right language model takes the left context, $t_1, t_2, ..., t_{i-1}$ into account for estimating the conditional probability. Language models are usually trained using large-scale unlabelled corpora. The conditional probabilities are most commonly learned using neural networks (Bengio et al., 2003, [161]), and the learned representations have been proven to be transferable to downstream natural language understanding tasks (Dai and Le, 2015, [162]); Ramachandran et al., 2016, [163]). Ramachandran et al. ([163]) extends Dai and Le ([162]) by proposing a pre-training method to improve the accuracy of sequence to sequence (seq2seq) models. The encoder and decoder of the seq2seq model is initialized with the pre-trained weights of two language models. To use ELMo in downstream tasks, the $(L + 1)$ - layer representations (including the global word embedding) for each token k are aggregated as:

$$ELMo_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j} \tag{3.2}$$

where s^{task} are layer-wise weights normalized by the softmax used to linearly combine the $(L+1)$-layer representations of the token k and γ^{task} is a task-specific constant. Given

a pre-trained ELMo, it is straightforward to incorporate it into a task-specific architecture for improving the performance. As most supervised models use global word representations x_k in their lowest layers, these representations can be concatenated with their corresponding context-dependent representations $ELMo_k^{task}$, obtaining $\left[\mathbf{x}_k; ELMo_k^{task}\right]$, before feeding them to higher layers. There are three main ways to use pre-trained contextual embeddings in downstream tasks ([160]):

1. Feature-based methods: One example of a feature-based is the method used by ELMo Peters et al. ([129]). Specifically, as shown in equation 3.2, ELMo freezes the weights of the pre-trained contextual embedding model and forms a linear combination of its internal representations. The linearly-combined representations are then used as features for task-specific architectures. The benefit of feature-based models is that they can use state-of-the-art handcrafted architectures for specific tasks. The RNN HG and RNN MHCA models that we described in Chapter 2 and that we used in the first part of this Chapter for our experiments implement ELMO representations (BERT representations have been obtained as well) to enhance metaphor detection task's performances.

2. Fine-tuning methods: starting with the weights of the pre-trained contextual embedding model, fine-tuning makes small adjustments to them in order to specialize them to a specific downstream task. One stream of work applies minimal changes to pre-trained models to take full advantage of their parameters. The most straightforward way is adding linear layers on top of the pre-trained models (Devlin et al., [155]; Lan et al., [164]). To apply pre-trained models to structurally different tasks, where task-specific architectures are used, as much of the model is initialized with pre-trained weights as possible. For instance, XLM (Lample and Conneau, [77]) applies two pre-trained monolingual language models to initialize the encoder and the decoder for machine translation, respectively, leaving only cross-attention weights randomly initialized.

3. Adapter methods: Adapters (Rebuffi et al., [165]) are small modules added between layers of pre-trained models to be trained in a multi-task learning setting. The parameters of the pre-trained model are fixed while tuning these adapter modules. Compared to previous work that fine-tunes a separate pre-trained model for each task, a model with shared adapters for all tasks often requires fewer parameters.

3.4 Quantitative Studies

3.4.1 Overall Performances

Before explaining all our experiments, we provide three tables (one for each dataset) with the overall obtained quantitative performances and scores (see figures 3.7, 3.8, 3.9).

MOH-X								
Embeddings Specifics					Metrics and Scores			
Temporal	Model	Main Corpus	Alignment	Slice	Precision	Recall	F1 Score	Accuracy
No	GloVe*	Wikipedia, Gigaword, Common Crawl*	NA	All	77.00%	81.00%	78.00%	79.00%
No	Word2Vec	Wikipedia	NA	All	**81.00%**	79.00%	80.00%	**81.00%**
Yes	Word2Vec	Full Coha SGNS	Procrustes	All	79.00%	81.00%	79.00%	80.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1900	79.00%	81.00%	80.00%	80.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1910	77.00%	83.00%	80.00%	79.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1920	79.00%	80.00%	79.00%	80.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1930	78.00%	81.00%	79.00%	79.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1940	79.00%	81.00%	80.00%	80.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1950	**81.00%**	80.00%	80.00%	**81.00%**
Yes	Word2Vec	Coha Word SGNS	Procrustes	1960	79.00%	80.00%	80.00%	80.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1970	80.00%	80.00%	80.00%	80.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1980	78.00%	81.00%	79.00%	80.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1990	78.00%	80.00%	79.00%	79.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	2000	80.00%	82.00%	**81.00%**	**81.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1900	79.00%	81.00%	80.00%	80.00%
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1950	77.00%	81.00%	79.00%	79.00%
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1990	76.00%	83.00%	79.00%	79.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1900	78.00%	81.00%	79.00%	80.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1950	80.00%	78.00%	79.00%	80.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1990	76.00%	**84.00%**	80.00%	79.00%
Yes	Word2Vec	Eng-fiction	Procrustes	1900	77.00%	82.00%	79.00%	79.00%
Yes	Word2Vec	Eng-fiction	Procrustes	1950	77.00%	82.00%	79.00%	79.00%
Yes	Word2Vec	Eng-fiction	Procrustes	1990	80.00%	81.00%	80.00%	**81.00%**
Yes	Word2Vec	Full CoHa Word CADE	CADE – Compass	All	**81.00%**	79.00%	79.00%	80.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1900	77.00%	80.00%	78.00%	79.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1910	78.00%	79.00%	79.00%	79.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1920	78.00%	80.00%	78.00%	79.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1930	79.00%	80.00%	79.00%	80.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1940	80.00%	78.00%	79.00%	80.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1950	79.00%	81.00%	80.00%	80.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1960	77.00%	81.00%	79.00%	79.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1970	80.00%	80.00%	80.00%	80.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1980	79.00%	79.00%	79.00%	79.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1990	80.00%	79.00%	79.00%	80.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	2000	78.00%	82.00%	79.00%	80.00%

Figure 3.7: Results related to MOH-X dataset, with every single embedding: recent temporal slices lead to better results.

3.4.2 RNN Models with HistWords and ELMO

These experiments have been performed using the **HistWords - SGNS temporal embeddings (English All, English-Fiction, CoHa Word, CoHa Lemma)**. We used the three main datasets that have been previously introduced: MOH-X, VUA and TroFi. We implemented the aforementioned embeddings inside the state of the art Recurrent Neural Network Hidden GloVe and Multi-Head Context Attention models, in conjunction with the ELMO vectors and replacing the standard GloVe embeddings. We already had the ELMO representations for each one of the three datasets, whereas we had to apply specific preprocessing and customizations respectively to the HistWords embeddings and the models' scripts. A few modifications had to be performed in order to run our experiments:

1. After downloading and collecting the HistWords - SGNS embeddings, their slices had to be made *gensim-compatible* (Word2Vec architecture), combining the numpy vectors and the vocabularies for each decade. The newly obtained representations for each decade have then been used as inputs for the models' embedding matrix, replacing GloVe vectors. A *Full CoHa Word SGNS* slice was also created, concatenating all *CoHa Word HistWords - SGNS* decade slices (in order to have a representation similar to the *Full CoHa Word CADE* one, which we will introduce later;

2. The code of the Recurrent Neural Network models' scripts had to be modified in order to accept the new embeddings, different from GloVe, and especially their new

VUA								
Embeddings Specifics					Metrics and Scores			
Temporal	Model	Main Corpus	Alignment	Slice	Precision	Recall	F1 Score	Accuracy
No	GloVe*	Wikipedia, Gigaword, Common Crawl*	NA	All	72.00%	**76.00%**	**74.00%**	93.00%
No	Word2Vec	Wikipedia	NA	All	75.00%	69.00%	72.00%	93.00%
Yes	Word2Vec	Full Coha SGNS	Procrustes	All	76.00%	71.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha Word SGNS	Procrustes	1900	76.00%	72.00%	**74.00%**	**94.00%**
Yes	Word2Vec	Coha Word SGNS	Procrustes	1950	76.00%	71.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha Word SGNS	Procrustes	1990	76.00%	71.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1900	**77.00%**	70.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1910	76.00%	71.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1920	76.00%	70.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1930	**77.00%**	70.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1940	**77.00%**	68.00%	72.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1950	**77.00%**	69.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1960	75.00%	73.00%	**74.00%**	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1970	76.00%	70.00%	73.00%	93.00%
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1980	76.00%	71.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1990	76.00%	71.00%	73.00%	**94.00%**
Yes	Word2Vec	Coha lemma SGNS	Procrustes	2000	76.00%	70.00%	73.00%	**94.00%**
Yes	Word2Vec	Eng-all SGNS	Procrustes	1900	**77.00%**	69.00%	73.00%	**94.00%**
Yes	Word2Vec	Eng-all SGNS	Procrustes	1950	74.00%	74.00%	**74.00%**	**94.00%**
Yes	Word2Vec	Eng-all SGNS	Procrustes	1990	**77.00%**	70.00%	73.00%	**94.00%**
Yes	Word2Vec	Eng-fiction	Procrustes	1900	75.00%	71.00%	73.00%	**94.00%**
Yes	Word2Vec	Eng-fiction	Procrustes	1950	75.00%	73.00%	**74.00%**	**94.00%**
Yes	Word2Vec	Eng-fiction	Procrustes	1990	76.00%	70.00%	73.00%	**94.00%**
Yes	Word2Vec	Full CoHa Word CADE	CADE – Compass	All	67.00%	72.00%	70.00%	92.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1900	76.00%	71.00%	**74.00%**	**94.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1910	75.00%	73.00%	**74.00%**	**94.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1920	72.00%	74.00%	73.00%	93.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1930	76.00%	70.00%	73.00%	**94.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1940	74.00%	74.00%	**74.00%**	**94.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1950	72.00%	74.00%	73.00%	93.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1960	75.00%	72.00%	73.00%	**94.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1970	75.00%	71.00%	73.00%	93.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1980	73.00%	75.00%	**74.00%**	93.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1990	75.00%	71.00%	73.00%	93.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	2000	76.00%	71.00%	73.00%	**94.00%**

Figure 3.8: Results related to VUAsequence dataset, with every single embedding: middle temporal slices lead to better results.

dimensions.

First Experiment: RNN HG and 1950 Decade Slice

The first test has been conducted using the Recurrent Neural Network Hidden GloVe model and the 1950 HistWords - SGNS decade slice. Results were then compared to the state of the art.

1. The results are generally better, compared to the state of the art, for all three main datasets, with the exception of TroFi as far as the *eng-fiction* slices are concerned;

2. With the exception of TroFi, all other HistWords embeddings registered an improvement of their precision scores (precision for VUA dataset increases to 76-77 % from 72 %). Thus, metaphor detection task's prediction scores are pretty much always higher when temporal embeddings are implemented. Recall scores often decrease in an inversely proportional manner compared to precision, and even though this could have been expected sometimes, it is peculiar to see such a pattern presenting itself so many times;

3. Overall, MOH-X is the best performing dataset with this specific decade slice: using *all-eng* and *coha-word* embeddings, precision, accuracy and F1 scores ended up being superior to the state of the art ones.

TROFI								
Embeddings Specifics					Metrics and Scores			
Temporal	Model	Main Corpus	Alignment	Slice	Precision	Recall	F1 Score	Accuracy
No	GloVe*	Wikipedia, Gigaword, Common Crawl*	NA	All	68.00%	76.00%	71.00%	74.00%
No	Word2Vec	Wikipedia	NA	All	70.00%	71.00%	71.00%	74.00%
Yes	Word2Vec	Full Coha SGNS	Procrustes	All	69.00%	73.00%	71.00%	74.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1900	69.00%	73.00%	71.00%	74.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1950	69.00%	74.00%	71.00%	74.00%
Yes	Word2Vec	Coha Word SGNS	Procrustes	1990	70.00%	72.00%	71.00%	74.00%
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1900	69.00%	73.00%	71.00%	74.00%
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1950	69.00%	73.00%	71.00%	74.00%
Yes	Word2Vec	Coha lemma SGNS	Procrustes	1990	70.00%	72.00%	71.00%	74.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1900	71.00%	71.00%	71.00%	**75.00%**
Yes	Word2Vec	Eng-all SGNS	Procrustes	1910	**72.00%**	70.00%	71.00%	**75.00%**
Yes	Word2Vec	Eng-all SGNS	Procrustes	1920	70.00%	72.00%	71.00%	74.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1930	70.00%	71.00%	71.00%	74.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1940	71.00%	71.00%	71.00%	**75.00%**
Yes	Word2Vec	Eng-all SGNS	Procrustes	1950	68.00%	75.00%	71.00%	74.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1960	69.00%	73.00%	71.00%	74.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1970	70.00%	72.00%	71.00%	74.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1980	71.00%	72.00%	71.00%	74.00%
Yes	Word2Vec	Eng-all SGNS	Procrustes	1990	70.00%	73.00%	71.00%	74.00%
Yes	Word2Vec	Eng-fiction	Procrustes	1900	69.00%	73.00%	71.00%	74.00%
Yes	Word2Vec	Eng-fiction	Procrustes	1950	68.00%	75.00%	71.00%	73.00%
Yes	Word2Vec	Eng-fiction	Procrustes	1990	70.00%	73.00%	71.00%	74.00%
Yes	Word2Vec	Full CoHa Word CADE	CADE – Compass	All	68.00%	**77.00%**	**72.00%**	74.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1900	70.00%	74.00%	**72.00%**	74.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1910	69.00%	76.00%	**72.00%**	**75.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1920	69.00%	76.00%	**72.00%**	**75.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1930	70.00%	74.00%	**72.00%**	**75.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1940	69.00%	75.00%	**72.00%**	74.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1950	69.00%	76.00%	**72.00%**	**75.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1960	69.00%	75.00%	**72.00%**	74.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1970	70.00%	74.00%	**72.00%**	**75.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1980	69.00%	76.00%	**72.00%**	74.00%
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	1990	70.00%	74.00%	**72.00%**	**75.00%**
Yes	Word2Vec	CoHa Corpus Slices	CADE – Compass	2000	69.00%	75.00%	**72.00%**	**75.00%**

Figure 3.9: Results related to TroFi dataset, with every single embedding: old temporal slices lead to results with a better balance of precision and recall, and a better accuracy. Precision increases with time at the price of recall, but not for eng-all SGNS where the best precision is observed in 1910 slices.

Observations

So far, *Eng-All* and *coha-word* slices seem to be the best performing ones for the temporal metaphor detection task. Accuracy and F1 scores are often higher than state of the art ones.

Second Experiment: RNN MHCA and 1950 Decade Slice

The second test has been performed using the same 1950 decade slice as before, but with the Recurrent Neural Network Multi-Head Context Attention model, in order to check whether its results could be as good as the RNN HG model's ones. Results have then been compared once again to the state of the art. Performances are worse than before and lower than state of the art (exception being made for a few cases in which we noticed an increase in precision scores).

Observations

Since the combination of RNN HG model and SGNS HistWords embeddings for 1950 decade slice gave better-than-the-state of the art scores and achieved better performances than the RNN MHCA model, further experiments have been conducted using the first of the two models.

Third Experiment: RNN HG and Slices Spanning the entire 20th Century

Firstly, also 1900 and 1990 decade slices of all four HistWords - SGNS embeddings have been tested, in order to confirm whether temporal embeddings achieve better performances than state of the art GloVe representations across multiple time spans. It was interesting to observe the results of all three datasets, checking whether their behavior changes in a way that could be linked to their temporal connotations.

1. Overall, MOH-X is still the best performing dataset, and it is worth noting that it behaves in different ways depending on which HistWords - SGNS embeddings' class is used. While *coha-word* and *eng-all* embeddings achieve better scores in correspondence of the median slice (1950), *coha-lemma* and *eng-fiction* achieve higher scores in correspondence of the extreme slices (1900 for *coha-lemma*, 1990 for *eng-fiction*). In order to understand whether *coha-word* and *eng-all* embeddings (the ones that performed better across the median temporal slice) might have detected some kind of loss of an initial metaphorical meaning of words towards the end of the 20th century, it is necessary to perform further experiments with the remaining decade slices as well;

2. TroFi dataset almost always presents higher precision scores in correspondence of the 1990 decade slice: this seems reasonable since we know that the data it is composed of was taken from the '87-'89 Wall Street Journal corpus. The temporal information contained in the embeddings lead to more accurate metaphor detection performance towards the decade slice of the same time period of the dataset. Among the other two decade slices, precision scores obtained in correspondence of the extreme 1900 slice is either the same or higher than the median ones, with the latter never achieving better scores than the extreme ones (as it happens for MOH-X).

3. VUA dataset behaves in a similar way to TroFi. This suggests that MOH-X could behave in a different way because of its sentences' unknown temporal connotations. *CoHa-Lemma* are the only HistWords - SGNS embeddings that provide higher precision scores when using the 1950 decade slice (compared to the 1990 one). Although the pattern is still not the same as the one of MOH-X (1990 slice does not provide lower precision scores than the mid-century one), the last experiments concerning VUA dataset have been performed using these embeddings.

Observations

It is indeed possible to say that so far temporal embeddings have enhanced the performance of metaphor detection task across different time spans, using the Recurrent Neural Network Hidden GloVe model. Results are good for all datasets: with HistWords - SGNS embeddings, precision scores of the predictions are almost always higher than the state of the art, and the same thing often happens for F1 and accuracy scores as well. The final tests were performed using all the remaining decades alices from 20th century and only the best performing embeddings for each dataset:

1. *CoHa-word* for MOH-X dataset;

2. *CoHa-lemma* for VUA dataset;

3. *Eng-all* for TroFi dataset.

We were able to affirm that:

1. **MOH-X** is still the best performing dataset overall, and it is worth noting that it behaves in a different way than VUA and TroFi. In fact, while metrics scores obtained with temporal word embeddings are generally better in terms of precision and accuracy compared to standard GloVe embeddings, recall in particular is generally always higher than precision (with a peak of 82% in correspondence of the 2000 temporal slice). In the other two datasets, instead, we can clearly see an opposite pattern, with an increase of precision scores and a parallel and constant drop of recall ones.

2. As far as the **VUA** dataset is concerned, precision scores are always much higher than state of the art, but at the expense of recall scores, that drop significantly. F1 and accuracy scores either increase or decrease without a clear pattern.

3. As far as the **TroFi** dataset is concerned, a stable increase of precision scores can be observed (even if it is not as huge as with VUA dataset), as well as an equally steady drop in recall scores. F1 and accuracy scores either increase or decrease without a clear pattern.

It is possible to say that temporal embeddings kept enhancing metaphor detection task performances, even across multiple time spans spanning the whole 20th century.

3.4.3 RNN Models with Wikipedia and ELMO

Finally, it is interesting to see whether it is indeed the temporal connotation of the SGNS embeddings that improve metaphor detection task performances, or just their Word2Vec architecture as opposed to the one of GloVe vectors. Therefore, previous results have been compared to those obtained using **a-temporal Word2Vec embeddings trained on the Wikipedia corpus**. We still used MOH-X, VUA and TroFi datasets. We only implemented the aforementioned embedding inside the state of the art Recurrent Neural Network Hidden GloVe, in conjunction with the ELMO vectors and replacing the standard GloVe embeddings. The Wikipedia embeddings were already in Word2Vec format, they did not require any preprocessing.

1. It can be noticed that for TroFi and VUA datasets, better results were still obtained using temporal SGNS embeddings compared to a-temporal Wikipedia representations. This is particularly interesting since these two datasets contain data with known and clear temporal connotations.

2. As far as MOH-X dataset is concerned, while RNN HG model achieves better results with Wikipedia Word2Vec embeddings than with GloVe, we can see that this is one of the few cases in which precision scores are higher than recall ones.

3. The only other two occurrences in which we observe precision scores as being higher than recall ones, for MOH-X dataset, have been registered with the 1950 decade slices of *coha-word* and *eng-all*. Temporal embeddings are therefore definitely more effective than GloVe for metaphor detection task: they also lead to better results when used with datasets whose temporal aspects are known.

3.4.4 Intermediate Results

We can affirm that temporal embeddings generally enhance Metaphor Detection task performances. Results are very good for all datasets: with HistWords - SGNS embeddings, precision scores are almost always higher than the state of the art, and lots of times the same thing ir registered for F1 score and accuracy too. TroFi dataset generally presents higher precision scores with the 1990 decade slice: this seems reasonable since the data it is composed of mainly comes from '88-'89 Wall Street Journal corpus. Overall, MOH-X is the best performing dataset, and it behaves in a particular way in the experiments with *all-eng* and *coha-word* slices. Here, precision scores are higher in correspondence of the median 1950 decade slice and lower in correspondence of the 1900 and 1990 ones. This could mean that for this specific dataset, an original metaphorical meaning of words got lost in time, leading to people using them in more common scenarios towards the end of the 20th century. Although Wikipedia embeddings are not trained on the Wikipedia corpus with specific focus on temporal aspects, they achieve results with higher precision scores than the state of the art. This suggests that the Word2Vec architecture is always superior to GloVe for metaphor detection task. TroFi and VUA datasets obtained better results with temporal embeddings, and this is particularly interesting thinking that these are the two datasets containing sentences whose temporal connotations are known. To summarize:

1. **Word2vec architecture (HistWords - SGNS embeddings) works better than GloVe architecture for metaphor detection**;
2. **HistWords - SGNS embeddings schieve better results than GloVe embeddings**;
3. **HistWords - SGNS temporal embeddings perform better on datasets with known temporal aspects (TroFi and VUA) compared to the atemporal Wikipedia embeddings**.

3.4.5 RNN Models and Compass-Aligned CoHa Slices

The following experiments have been performed using word embeddings obtained by aligning all different decades slices of the CoHa corpus (ranging from 1820 to 2000) with *CADE - Compass* alignment method. These representations, as in previous experiments, replaced the standard GloVe embeddings. We always used MOH-X, VUA and TroFi datasets. We implemented the aforementioned embeddings inside the state of the art Recurrent Neural Network Hidden Glove model. All results have been analyzed and commented, and they can be consulted looking at the final tables 3.7, 3.8 and 3.9. Slices of the CoHa corpus for each decade needed to be aligned with CADE - Compass in order to perform equivalent experiments to the ones with other embeddings. Specific preprocessing steps had to be applied before aligning with Compass. The CoHa corpus in text format is composed by one main folder for each decade ranging from 1820 to 2000 and containing all text files for each one of those decades. The first step consisted in concatenating all text files for each folder, obtaining a final corpus for each decade. The pre-processing procedures applied to clean all the resulting corpora were:

1. Stripping HTML tags;
2. Removing text between square brackets;

3. Replacing all contractions;

4. Removing stopwords (identified using the *nltk Python* library;

5. Cleaning the resulting text through several regex expressions, such as lower-case.

At this point, the alignment using CADE and Compass could be performed through the following steps:

1. Creating the main compass file by concatenating all the processed CoHa decade slices;

2. Training the obtained compass;

3. Training all the different slices from the compass obtaining their respective models;

4. Converting the CoHA compass models in Word2Vec format, so that they could have the same architecture of the HistWords - SGNS and Wikipedia embeddings and be used inside the modified Recurrent Neural Network model.

Finally, only the aligned models of the decades slices ranging from 1900 to 2000 were kept, so that the results could be comparable to the previous ones. A *Full CoHa Word CADE* model was also obtained by training the compass on all the aforementioned decade slices: this will be used in Chapter 4 for qualitative analyses as well.

3.4.6 Intermediate Results

The results were very similar to the ones obtained with other Word2Vec embeddings (HistWords - SGNS and Wikipedia). Once again, we can confirm that this architecture is more effective than GloVe for metaphor detection tasks. MOH-X is always the best performing dataset, with way higher-than-state-of-the-art scores. The scores obtained with VUA dataset are basically on par with the ones achieved by Wikipedia and HistWords - SGNS embeddings, even if the latter provided higher precision scores across specific decades slices. We can therefore affirm that representations obtained through Procrustes alignment seem to impact British National Corpus data more than Compass aligned embeddings. As far as TroFi dataset is concerned instead, we noticed that Compass-aligned representations performed generally better than Procrustes aligned HistWords embeddings. Although precision scores were sometimes lower, recall and F1 scores were visibly better. In particular, we registered a steady improvement of F1 scores across all decades slices, moving from a permanent 71% to 72%. In this case, Compass alignment method seemed to impact Wall Street Journal data more than Procrustes alignment.

3.4.7 Conclusions

Combining the observations gathered from all the performed experiments, these are our conclusions:

1. **Word2vec architecture (HistWords - SGNS, CoHa and Wikipedia embeddings) works better than GloVe architecture for metaphor detection**;

2. **HistWords - SGNS embeddings achieve better results than GloVe**;

3. **HistWords - SGNS temporal embeddings perform better on the datasets with known temporal connotations (TroFi and VUA) compared to the a-temporal Wikipedia embeddings**;

4. **MOH-X dataset is generally the one that achieves better results with the different types of word representations**;

5. **Procrustes alignment (HistWords - SGNS) and CADE - Compass alignment methods (CoHa corpus) lead to similar performances and results. Although, while the latter performs better on the TroFi dataset (data extracted from Wall Street Journal corpus), the first one impacts slightly more VUA dataset (data extracted from the British National Corpus)**.

3.5 Qualitative Analysis and Evaluation

3.5.1 Background and Motivational Questions

At this point, we took a more qualitative look at the predictions made by the RNN HG model with the different types of embeddings used so far, analyzing the characteristics of the actual words that were correctly or mistakenly identified as metaphors. Since the range of our previous experiments was too big and it was not feasible to inspect all the results, we decided to focus on MOH-X, VUA and TroFi datasets' predictions obtained with 4 specific embeddings (pictures 3.7, 3.8 and 3.9):

1. Full CoHa Word CADE;

2. GloVe (state of the art representation);

3. CoHa Word CADE 1990 Decade Slice;

4. CoHa Word HistWords SGNS 1990 Decade Slice.

The first two representations are trained on full corpora (in particular the *Full CoHa CADE* one spans the 19th and the 20th centuries). The last two representations, instead, were obtained from a specific slice. We chose the 1990 decade slice because of VUA and TroFi datasets' sentences temporal connotations, because this slice generally provided good quantitative performances, and to make the qualitative analyses of these two embeddings comparable. Therefore, the usage of these 4 corpora allowed us to look at predictions made by the RNN HG model with both temporal and a-temporal embeddings, and with both full and singular corpora slices. In order to check all the actual predictions made by our model for MOH-X and TroFi, we had to combine each one of their 10-folds intermediate results, since these two datasets are not split into train, validation and test sets like VUA. For each one of the three state of the art datasets, we checked:

1. **Correctly identified metaphors**;

2. **Mistakenly identified metaphors**.

The different focuses of the qualitative analysis for the three datasets were the following:

1. **MOH-X**: **verbs** (a column for nouns is also given in the dataset, but they are not included in the evaluation labels);

2. **VUA**: most frequent patterns of **parts-of-speech**' positions in the sentences and **genres** of the sentences coming from the British National Corpus;

3. **TroFi**: **verbs**.

To sum up, the questions that we wanted to answer are the following:

1. What are the characteristics of correctly and mistakenly metaphorical predictions obtained through our temporal metaphor detection experiments? Could it be that parts-of-speech tokens contained in sentences concerning a particular topic get correctly identified with higher precision and/or frequency compared to others? Are specific language genres (this question is particularly related to VUA dataset) easier to correctly classify as metaphors than others?

2. Are there any patterns concerning the words that are most similar (nearest neighbors) to the most frequently identified metaphorical words (nouns, verbs, and so on?

3.5.2 Language Analysis with Word2Vec

Word2Vec provides very useful methods to perform language-related analyses. In this chapter we analyzed the actual words that were correctly or mistakenly predicted as metaphors by the RNN-based models. Thanks to Word2Vec methods, we have then been able to observe the **nearest neighbors** (the most similar words) of the predictions obtained with the various embeddings, and see whether they could lead us to confirm previously detected patterns, or discover new ones altogether. These analyses were mostly performed to check MOH-X and TroFi predictions, since they provide precise information about metaphorical verbs and nouns. Through word embeddings' visualizations it is also possible to see that words belonging to similar contexts and having similar meanings are characterized by closer representations in the vector space (an example can be observed in figure 3.10).

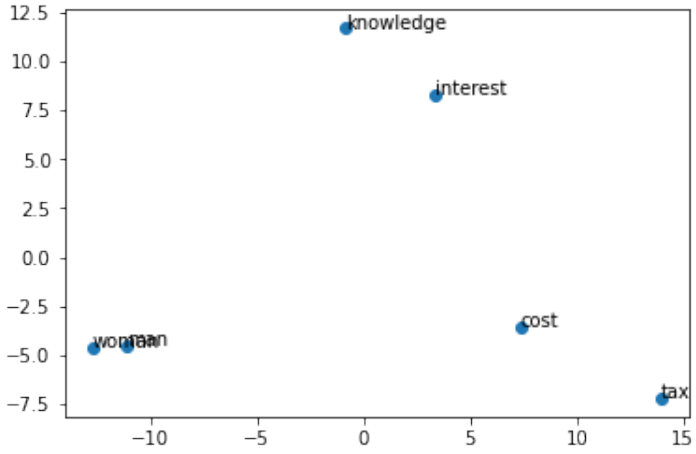

Figure 3.10: Visual representations of words in the vector space.

Some other useful measures for this kind of tasks are the *Cosine distance*, which computes the distance between 1-D arrays, and the *Cosine similarity*. The Cosine distance between two vectors u and v, is defined as:

$$1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2} \tag{3.3}$$

where $u \cdot v$ is the dot product of u and v ([101]), and the parameters are as follows:

1. *u(N,)* array_like - The first input array;
2. *v(N,)* array_like - The second input array;
3. *w(N,)* array_like, optional - The weights for each value in u and v. Default value is 'None', which gives each value a weight of 1.0.

The returned value is the Cosine distance between the two vectors. When this value is close to 0, it means that the two observed vectors are more different from each other (there is no match between them since they are orthogonal to each other); instead, if the cosine distance is close to 1, it means that the vectors are more similar to each other.

The Cosine similarity is obtained as follows:

$$cosine\_similarity = 1 - cosine\_distance. \tag{3.4}$$

3.5.3 Full CoHa Word CADE

MOH-X Predictions

Correct Metaphorical Predictions Looking at the metaphorical verbs that were correctly predicted as such, we can see that the most frequently identified ones are the following:

1. **absorb** - 5 occurrences;
2. **swallow** - 5 occurrences;
3. **drift** - 4 occurrences;
4. **precipitate** - 3 occurrences.

These verbs were always correctly classified. Let's check some of the sentences in which these verbs appear:

1. *He **absorbed** the knowledge or beliefs of his tribe*;
2. *He **absorbed** the costs for the accident*;
3. The immigrants were quickly **absorbed** into society;
4. *The Nazis **swallowed** the Baltic countries*;
5. She **swallowed** the last words of her speech;
6. *I **swallowed** my anger and kept quiet*;
7. *Stock prices are **drifting** higher*;

8. *My son **drifted** around for years in California before going to law school*;
9. *The crisis **precipitated** by Russia's revolution.*;
10. *Our economy **precipitated** into complete ruin..*

As we can notice, each one of these verbs is used in very different contexts (knowledge, economics, society, history/politics, everyday expressions). In particular, **economical/political** and **emotional/feelings** related connotations are recurrent: **although using a different set of test data, we already noticed these topics' patterns in the previous transfer learning experiments**. Even if nouns are not counted as metaphorical targets within the evaluation labels, we can notice that the most frequently recurring ones among the correct predictions are:

1. **people** - 5 occurrences;
2. **market** - 4 occurrences;
3. **economy** - 4 occurrences;
4. **feeling** - 4 occurrences;
5. **story** - 4 occurrences.

These nouns are related to the contexts that we analyzed before. Wrong Metaphorical Predictions The words that were mistakenly classified as metaphors were just 29, therefore in this case the analysis is pretty straightforward. Among the remaining verbs and nouns, we can find only three occurrences of words related to the topics and contexts related to the correctly classified metaphors that we saw before, respectively:

1. **buy** - verb, 1 occurrence, related to *economics*;
2. **book** - noun, 1 occurrence, related to *knowledge*;
3. **love** - noun, 1 occurrence, related to *emotions* and *feelings*.

Let's have a look at the only sentence in which *buy* appears:

- *She wanted to **buy** his love with her dedication to him and his work.*

Interestingly enough, it can be observed that even if the original label for this verb's occurrence was classified by the annotators as literal, it is a very fine line that we walk here. Someone could actually consider this case as a metaphorical one. Besides, there is a very interesting mix of **economical** and **emotional** contexts. The other 26 words that were mistakenly classified as metaphors do not share a common context.

VUA Predictions

Correct Metaphorical Predictions The words that have been correctly classified as metaphorical with higher frequency belong to the *conversation* domain in British National Corpus. *Fiction* is the second one followed by *news*. It is pretty interesting to observe that only one sentence belonging to the *Academic* genre has been correctly classified:

- *As Sartre had **put** it **in** What is Literature?.*

whose words are characterized by the following *pos-seq* sequence:

- ['ADP', 'PROPN', 'VERB', **'VERB'**, 'PRON', **'ADP'**, 'NOUN', 'VERB', 'NOUN', 'PUNCT', 'PUNCT']

It seems like the model is not very good at identifying all the metaphorical words inside sentences that start with *ADP* (adposition) pos tags, such as *'as'* in this specific case. As we can see in figure 3.11, among the only pos-seq tags sequences combinations that appear with a frequency equal to or higher than 1%, the verb is always placed at the second position right after a *PROPN* (proper noun), a *PRON* (pronoun), or *NOUN*.

4%	['PRON', 'VERB', 'PUNCT']
1%	['PROPN', 'VERB', 'PUNCT']
1%	['PRON', 'VERB', 'PRON', 'PUNCT']
1%	['NOUN', 'PRON', 'VERB', 'ADV', 'VERB', 'PUNCT']
1%	['PRON', 'VERB', 'DET', 'NOUN', 'PUNCT']
1%	['PROPN', 'VERB', 'VERB', 'PUNCT']
<1%	['ADV', 'PRON', 'VERB', 'PRON', 'PUNCT']
90%	(Other)

Figure 3.11: Most frequent pos-seq tags sequences' combinations of correctly classified VUA metaphorical sentences - Full CoHa Word CADE.

Wrong Metaphorical Predictions As far as the wrong predictions are concerned, the order of the most frequent British National Corpus genres is the same as before, but this time the *academic* one is not limited to one single sentence and instead has almost the same number of sentences of *fiction*. Looking at the predictions, we noticed that 54 sentences out of 280 are actually questions, and almost all of them belong to the two most frequent genres (*conversation* and *fiction*): this could mean that the model finds it harder to correctly detect metaphors inside questions. The pos-seq tags sequences' combinations are not distinct enough to justify a specific analysis.

TroFi Predictions

Correct Metaphorical Predictions

At first sight, it could seem that the meanings of these verbs are related to different topics compared to those of MOH-X dataset (figure 3.12). In fact, a lot of these verbs apparently have **physical** connotations (*strike, attack, kill*). Analyzing the sentences more in detail, it appears that their metaphorical nature is caused by the use of the aforementioned verbs in **economical** and **emotional** contexts. Let's see a few examples:

1. *Inflation,* ***attacked*** *in the abortive Austral Plan of 1985 and now the target of new economic measures, has exceeded 300% over the past year*;
2. *"We're not* ***attacking*** *the core assets; we're looking at what we consider to be our less - profitable assets," the spokesman said*;
3. *The idea of having one child has* ***struck*** *a deeper root and less people are punished for violating the birth control plan...*;

TOP CATEGORIES		
lend	110	4%
step	104	4%
miss	98	4%
examine	98	4%
fly	93	4%
strike	92	3%
attack	90	3%
kill	78	3%
roll	75	3%
fill	75	3%
stick	74	3%
escape	71	3%
absorb	71	3%
destroy	69	3%
target	67	3%
drink	62	2%
plant	62	2%
(Other)	1,245	47%
ALL	2,634	100%

Figure 3.12: TroFi top correctly identified verbs - Full CoHa Word CADE.

4. *The ad **struck** a nerve in a public made more jittery by the obsessed mob that , in Sen. Goldwater 's name , hooted and jeered former President Eisenhower at the Republican National Convention that summer.*

In the first two examples, *attack* has been used to express concepts related to economical subjects; in the last two examples, *strike* has been used with a clear emotional connotation. Wrong Metaphorical Predictions Some of the verbs are the same that we found in the correct predictions(*attack, strike, kill*): this time, in most cases the model assigned a metaphorical meaning to these verbs when in reality they were just being used within their literal connotation (physical meaning). A few examples are the following:

1. *They claim that colon cancer , which **strikes** more than 100000 Americans a year, occurs only in people who inherit a particular gene from their parents that predisposes them to the cancer* - mistakenly classified as metaphor;

2. *POLISH LABOR UNREST SPREAD as thousands **struck** for higher wages* - mistakenly classified as metaphor;

3. *Col . North again proposed luring Col . Gadhafi to a spot where bombers could **target** him* - mistakenly classified as metaphor;

4. *Stones are laid close together , generally slanting inward , with each stone **touching** as many others as possible* - mistakenly classified as metaphor.

Nearest Neighbors Analysis

Searching for the nearest neighbors of one of the most frequently occurring **MOH-X** verbs, **swallow**, we noticed that one of the results, *suck*, appears in sentences alongside different frequently identified nouns (or nearest neighbors of the latter). An example:

- *The current boom in the **economy sucked** many **workers** in from abroad.*

As we can see, the topic of **economics** appears once again to be recurrent among metaphorical predictions. Besides, *suck* and other similar verbs have **physical connotations, but they assume figurative meanings when used within the economical context**, as we previously saw in the TroFi predictions. It is also interesting to see that nearest neighbors of frequently identified nouns, such as *industry* for *economy*, were found in sentences alongside correctly classified metaphorical verbs:

- ***Industry*** *will* ***stagnate*** *if we do not stimulate our* ***economy****.*

Stagnate also has **emotional** connotations, although negative, confirming an other previously observed pattern and linking **economics** to **feelings**. As far as **TroFi** predictions are concerned, among the words that are most similar to one of the most frequently correctly classified metaphorical verbs, *strike*, we find *strikes*, which can be both a verb and a noun. If it is used as a verb, it **originally has a physical connotation, but in various examples it assumes metaphorical meaning inside an *economical/political* contexts**, like in the following sentence:

- *The item veto* ***strikes*** *me as a good idea whose time has come, along with others common in our state constitutions.*

Observations

To summarize, for the three datasets:

1. **MOH-X**: Each one of the **correctly classified** verbs is used in very different contexts (knowledge, economics, society, history/politics, everyday expressions). In particular, **economical/political** and **emotional/feelings** related connotations are recurrent. The most frequently recurring nouns are related to the same aforementioned contexts. In the **wrong predictions**, only a few words related to those topics are found, and errors are sometimes related to ambiguous original annotations. Thanks to **nearest neighbors analysis**, we discovered new cases in which **verbs with physical connotations** assume figurative meanings when used within **economical and emotional contexts**, and **nouns related to economics** are found alongside correctly classified metaphorical verbs in the same topics.

2. **VUA**: The words that have been correctly classified as metaphorical with higher frequency belong to the **conversation** British National Corpus genre. Only one sentence belonging to the **academic** genre has been correctly classified. *PROPN* (proper noun), *PRON* (pronoun), or *NOUN* followed by a *VERB* seem to be the most frequent combinations of pos-seq tags sequences in the predictions. The model could find it harder to correctly detect metaphors inside questions.

3. **TroFi**: The most frequently identified verbs have literal meanings characterized by **physical** connotations; anyway, they assume figurative meanings when used in contexts such as **economics** and **emotions**. Metaphorical labels are often erroneously assigned to literal words due to the original dataset's ambiguous annotations. As far as these predictions are concerned, the nearest neighbors analysis performed on the *Full CoHa CADE* embedding allowed us to discover new sentences in which **physical verbs assume figurative meaning when used in economical/political contexts**.

3.5.4 GloVe

MOH-X Predictions

Correct Metaphorical Predictions Looking at the metaphorical verbs that were correctly predicted as such, we can see that the most frequently identified ones are the same as with *Full CoHa Word CADE*:

1. **absorb** - 5 occurrences;
2. **swallow** - 5 occurrences;
3. **drift** - 4 occurrences;
4. **erupt** - 3 occurrences;
5. **precipitate** - 3 occurrences.

These verbs have been once again correctly classified in each one of their occurrences. Let's see the sentences in which the new verb that we didn't encounter in the previous analysis (*erupts*) appears:

1. *Unrest* ***erupted*** *in the country.*;
2. *The tooth* ***erupted*** *and had to be extracted.*;
3. A rash **erupted** on her arms after she had touched the exotic plant..

In the first example, the verb acquires metaphorical meaning in an **emotional context**, confirming the link between metaphors and feelings. As we previously noticed, each one of the other verbs is used in other contexts such as knowledge, economics, society, history/politics, everyday expressions). In particular, **economical/political** and **emotional/feelings** related connotations are recurrent. The most frequently recurring nouns among the *GloVe* correct predictions are always related to these contexts:

1. **people** - 5 occurrences;
2. **word** - 4 occurrences;
3. **market** - 4 occurrences;
4. **idea** - 4 occurrences;
5. **story** - 4 occurrences;
6. **feeling** - 4 occurrences.

Wrong Metaphorical Predictions The MOH-X words that were mistakenly classified as metaphors were just 11. These wrong predictions are even less than the ones with *Full CoHa CADE*, and among them we can find only three words related to the topics and contexts of the correctly classified metaphors that we saw before, respectively:

1. **melt** - verb, 1 occurrence, which could acquire figurative meaning in an *emotional* context;
2. **book** - noun, 1 occurrence, related to *knowledge*;

3. **finding** - noun, 1 occurrence, related to *knowledge*.

Let's see the sentence in which *melt* appears:

1. *The wax **melted** in the sun.*

Apparently, the model learnt pretty well that this verb has been more and more acquiring a figurative meaning, and it erroneously classified it as a metaphor.

VUA Predictions

Correct Metaphorical Predictions The words that have been correctly classified as metaphors with higher frequency belong to the *conversation* British National Corpus genre. The order for the other three genres is *fiction*, then *news* and finally *academic*. This time there have been definitely more correctly classified words contained in sentences of the *academic* genre (and thus related to **knowledge** topics). Let's see a few practical examples:

1. While such notions may all **contains** some **elements** of truth , they are by no means complete explanations of criminal behavior.;
2. ***This** chapter will **examine** a **range** of theories which **attempt** to explain such behavior.*;
3. *Most people **have** their own notions or theories as to what causes criminal behavior.*.

Among the pos-seq tags sequences combinations of these predictions, the most frequent ones are respectively:

1. ['PRON', 'VERB', 'PUNCT'];
2. ['PROPN', 'VERB', 'PUNCT'].

The same thing happened for the *Full CoHa CADE* VUA correct predictions.

Wrong Metaphorical Predictions As far as the wrong predictions are concerned, the order of the most frequent British National Corpus genres is different this time, as we can see in figure 3.13:

Let's have a look at a few *conversation* sentences to try to understand what the classification problem is due to:

1. *You can **make** it different like and **make** it sort of still a dance area* - the model identified the two highlighted verbs as metaphors, while the sentence was originally labeled as entirely literal;
2. *You ought to **save** them wooden type of things for it* - the model classified the highlighted verb as metaphor, while it should have identified the noun **things** as such;
3. *How do you **get to that**?* - the model classified all the highlighted parts of speech as metaphors, while it should have identified only **that**.

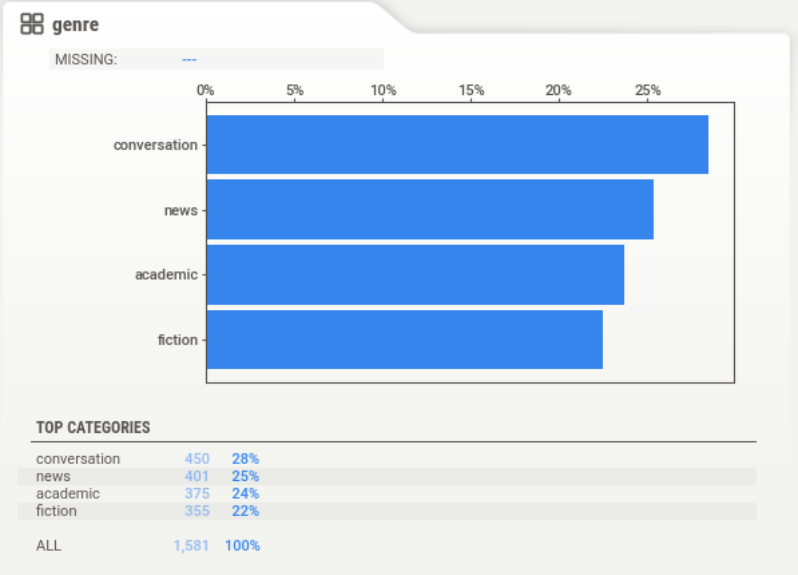

Figure 3.13: VUA top correctly identified genres' distribution - GloVe.

At least in the second and third cases, it seems like the errors in the predictions are not particularly huge: the model managed to recognize the parts of the sentences which contained figurative expressions, but the annotators chose to label just some of those words. These kind of *conversation* sentences seem to be mostly taken out of context and sometimes annotations could have been made differently, therefore these aspects could have something to do with the errors. The pos-seq tags sequences' combinations are not distinct enough to justify a specific analysis.

TroFi Predictions

Correct Metaphorical Predictions The verbs that we see in these correctly made predictions (figure 3.14) are more or less the same as the ones we found with *CoHa Full CADE* embedding (figure 3.12).

A lot of these verbs have **physical** connotations (*strike, attack, kill*). Analyzing the sentences more in detail, it appears that their metaphorical nature is caused by the use of the aforementioned verbs in **economical** and **emotional** contexts. Let's see a few examples:

1. *Thomas McGreevy, president of the New York fathers' rights group, adds that men don't consider being* ***struck*** *as spouse abuse*;

2. *Still , market participants say the firmer U.S. interest rates will* ***lend*** *support to the dollar if tomorrow 's report shows that the trade gap widened in June* ;

3. *"The U.S. could* ***miss*** *the boat if they play politics at the ADB and do n't put up the bucks, " says an ADB executive who declined to be identified .*

In all the three sentences, the verbs have acquired a figurative meaning because of the **emotional** contexts in which they have been used. Wrong Metaphorical Predictions

TOP CATEGORIES		
strike	122	4%
step	120	4%
lend	117	4%
miss	116	4%
attack	115	4%
fly	112	3%
examine	109	3%
fill	105	3%
stick	103	3%
kill	95	3%
destroy	90	3%
roll	89	3%
escape	80	2%
absorb	79	2%
touch	74	2%
cool	74	2%
drink	73	2%
(Other)	1,530	48%
ALL	3,203	100%

Figure 3.14: TroFi top correctly identified verbs - GloVe.

Some of the verbs are the same that we found in the correct predictions. In most cases the model assigned a metaphorical meaning to these verbs when in reality they were just being used within their literal connotation. A few examples are the following:

1. *Programs **targeted** for "possible privatization" include some Postal Service operations and the management of minimum - security prisons* - mistakenly classified as metaphor;

2. *In the past month, the company's shares have been **bouncing** around in the mid - to - high $50s and **touched** $60* - **touched** was correctly identified as metaphor, while **bouncing** should have been detected as literal;

3. *"They've **stuck** to the real nuts and bolts of their business."* - mistakenly classified as metaphor;

4. *For now, he **rides** around his opponents like a modern Jeb Stuart, staging lightning raids before returning to safety in the South* - mistakenly classified as metaphor.

Nearest Neighbors Analysis

In order to perform nearest neighbors' analysis with GloVe embeddings, we used Gensim's *glove2wordvec* method to load the representations in word2vec format. Thought-provoking discoveries emerged by our investigation on the **MOH-X** predictions. Searching for the nearest neighbors of one of the most frequently occurring nouns, **market**, we noticed that one of the results, *sales* (and other similar nouns such as *salesman*), appears in several sentences alongside different correctly classified metaphorical verbs:

1. *The **sales** tax is **absorbed** into the state income tax*;

2. ***Sales** were **climbing** after prices were lowered*;

3. *The **salesman** is aggressively **pushing** the new computer model.*

As we can see by these examples, the topic of **economics** appears once again to be recurrent among metaphorical predictions. Besides, we found new cases in which **verbs with physical connotations (*climb*, *push*) assume figurative meanings when used within the economical context**, as we previously saw in *CoHa CADE Full* TroFi predictions. The nearest neighbors of another frequently identified noun, *feeling*, confirmed another pattern that before had mainly emerged from **TroFi** predictions. Nouns such as *mind* were found in sentences whose metaphorical meaning was given by **verbs with a physical connotation used in emotional contexts**:

1. *Fear* ***clogged*** *her* ***mind***;
2. *His mind* ***groped*** *to make the connection*;
3. ***poison*** *someone's* ***mind***.

As we can see, *clog*, *grope* and *poison* originally have physical meanings, but they assumed a figurative one in **emotional** contexts. As far as **TroFi** predictions are concerned, even in this case the neighbor analysis revealed interesting patterns. For example, among the words that are most similar to the most frequently correctly classified metaphorical verb, *kill*, we can find the noun *hell*. This word is used alongside a **verb with an originally physical connotation (*stick*) that assumes metaphorical meaning inside an *emotional* context** in the following sentence:

- *The trick is to focus narrowly on the bonds among guys who have been through* ***hell*** *together and will* ***stick*** *together unto death , without bothering with their attitudes toward the cause for which they are fighting.*

This is a fascinating mix of almost all the different patterns related to verbs, nouns and related topics that we have encountered so far.

Observations

To summarize, for the three datasets:

1. **MOH-X**: We can notice patterns that are similar to the ones observed with the previous embedding. Metaphorical labels are sometimes wrongly assigned to literal verbs that are effectively used in a figurative way lots of times. Thanks to **nearest neighbors analysis**, we found new cases in which **verbs with physical connotations** assume figurative meanings when used within **economical and emotional contexts**, as we previously saw in *CoHa CADE Full* TroFi predictions and in these *GloVe* predictions as well.
2. **VUA**: The words that have been correctly classified as metaphorical with higher frequency belong to the **conversation** British National Corpus genre. This time way more **academic** sentences have been correctly classified, and in the **wrong predictions** the order of the genres is for the first time different. *PROPN* (proper noun), *PRON* (pronoun), or *NOUN* followed by a *VERB* are once again the most frequent combinations of pos-seq tags sequences in the predictions. Several times parts of the sentences which contained figurative expressions are successfully recognized, but since the annotators chose to label just some of these specific words, the predictions as a whole are wrong.

3. **TroFi**: The most frequently identified verbs have literal meanings characterized by **physical** connotations; anyway, they assume figurative meanings when used in contexts such as **economics** and **emotions**. Metaphorical labels are often erroneously assigned to literal words due to the original dataset's ambiguous annotations. As far as these predictions are concerned, the nearest neighbors analysis performed on the *GloVe* embedding allowed us to discover new sentences in which **physical verbs assume figurative meaning when used in emotional contexts**.

3.5.5 CoHa Word CADE 1990 Slice

MOH-X Predictions

Correct Metaphorical Predictions Looking at the metaphorical verbs that were correctly predicted as such, we can see that the most frequently identified ones are the same as before:

1. **absorb** - 5 occurrences;
2. **swallow** - 5 occurrences;
3. **drift** - 4 occurrences;
4. **erupt** - 3 occurrences,

These verbs have been once again correctly classified in each one of their occurrences. The most frequently recurring nouns among these correct predictions are related to the usual figurative contexts/topics that we have already encountered (see figure 3.15):

1. **people** - 5 occurrences;
2. **economy** - 4 occurrences;
3. **idea** - 4 occurrences;
4. **feeling** - 4 occurrences;
5. **market** - 4 occurrences;
6. **story** - 4 occurrences.

5	2%	people
4	1%	economy
4	1%	idea
4	1%	feeling
4	1%	market
4	1%	story
3	1%	interest
264	90%	(Other)

Figure 3.15: MOH-X nouns that most frequently appear among CoHa Word CADE slice's correct predictions.

Wrong Metaphorical Predictions The MOH-X words that were mistakenly classified as metaphors were just 16. Among them we can find only two words related to the topics and contexts of the correctly classified metaphors, respectively:

1. **solution** - noun, 1 occurrence, related to *knowledge*;
2. **problem** - noun, 1 occurrence, related to *knowledge*.

VUA Predictions

Correct Metaphorical Predictions The words that have been correctly classified as metaphors with higher frequency belong to the *conversation* British National Corpus genre. The order for the other three genres is *fiction*, then *news* and finally *academic*, confirming the pattern that we already saw with *Full CoHa CADE* embedding (figure 3.16):

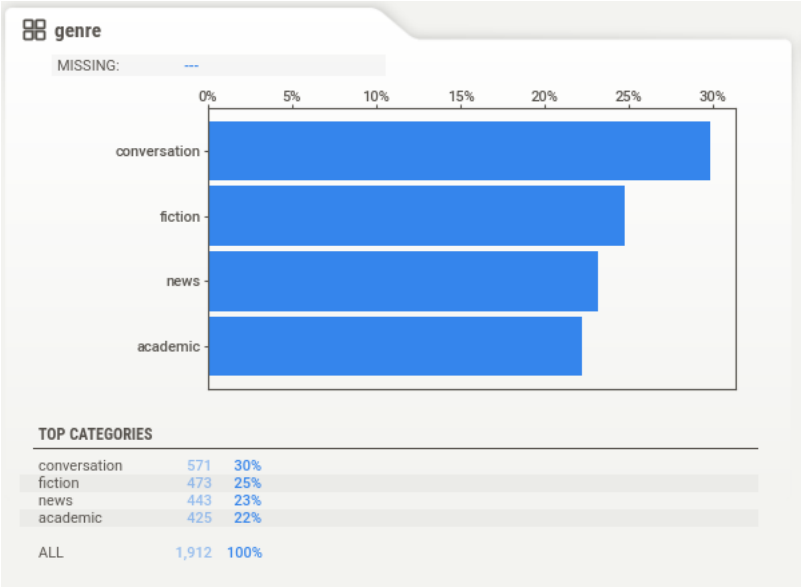

Figure 3.16: VUA genres' distribution for correct and wrong predictions alike - Full CoHa CADE and CoHa Word CADE 1990.

Among the pos-seq tags sequences combinations of these predictions, the most frequent ones are respectively:

1. ['PRON', 'VERB', 'PUNCT'];
2. ['PROPN', 'VERB', 'PUNCT'].

The same thing happened for the *Full CoHa CADE* and *GloVe* predictions.

Wrong Metaphorical Predictions As far as the wrong predictions are concerned, the order of the most frequent British National Corpus genres is the same as before. Once again, it seems like the model managed to recognize the parts of the sentences which contained figurative expressions, but the annotators chose to label just some of those words leading to erroneous predictions. The pos-seq tags sequences' combinations are not distinct enough to justify a specific analysis.

TroFi Predictions

Correct Metaphorical Predictions The verbs that we see in these correctly made predictions are the same as the ones we found with *CoHa Full CADE* and *Glove* embeddings (figures 3.12 and 3.14). A lot of these verbs have **physical** connotations (*strike, attack, kill*), and their metaphorical nature is once again determined by their use in **economical** and **emotional** contexts. Wrong Metaphorical Predictions Some of the verbs are the same that we found in the correct predictions. In most cases the model assigned a metaphorical meaning to these verbs when in reality they were just being used within their literal connotation. A few examples are the following:

1. *And it didn't help that law - enforcement officials have reported problems with 19- and 20-year - olds coming to Wyoming from neighboring states to **drink*** - mistakenly classified as metaphor;

2. *Its theme – Peter **dragging** Russia out of the dank Dark Ages – is vast and lofty, and perhaps only a man who dreamed and **drank** too much would have had the gumption to cut so wide and deep a swath into the heart of Russia* - **drank** was correctly identified as literal, while **dragging** which was also literal has been detected as metaphor;

3. *"He just tried to **eat** the propeller, " says Mr. Fox* - mistakenly classified as metaphor.

Nearest Neighbors Analysis

Nearest neighbors analysis on *CoHa Word CADE 1990 Slice* through Word2Vec lead to results and observations very similar to the ones obtained with *Full CoHa CADE* embeddings.

Observations

To summarize, for the three datasets:

1. **MOH-X**: We can notice patterns that are similar to the ones observed with the previous embeddings, especially to the ones obtained with *Full CoHa CADE* representations (the same thing can be said for nearest neighbors analysis).

2. **VUA**: The order of the British National Corpus' genres is the same of the *Full CoHa CADE* case. *PROPN* (proper noun), *PRON* (pronoun), or *NOUN* followed by a *VERB* are always the most frequent combinations of pos-seq tags sequences in the predictions. Even in this case, some equivocal annotations lead to words being erroneously given a metaphorical label.

3. **TroFi**: We can notice patterns that are similar to the ones observed with the previous embeddings, especially with the *Full CoHa Cade* one (the same thing can be said for nearest neighbors analysis).

3.5.6 CoHa Word HistWords/SGNS 1990 Slice

MOH-X Predictions

Correct Metaphorical Predictions We can notice a few new verbs besides the ones we have been getting used to (figure 3.17):

4	2%	swallow
4	2%	absorb
3	1%	lock
3	1%	climb
3	1%	steamroller
3	1%	inflate
3	1%	attack

Figure 3.17: MOH-X verbs that most frequently appear among CoHa SGNS 1990 slice's correct predictions.

Just like with TroFi in the previous analyses, some of these verbs have more physical connotations. Let's have a look at some of the sentences in which they have been identified:

1. *He* ***locked*** *his hands around her neck*;
2. *He was* ***locked*** *in a laughing fit*;
3. *Sales were* ***climbing*** *after prices were lowered*;
4. *The path* ***climbed*** *all the way to the top of the hill*;
5. *The new teacher tends to* ***steamroller***;
6. ***steamroller*** *the opposition*;
7. *The editors of the left - leaning paper* ***attacked*** *the new House Speaker*;
8. *I* ***attacked*** *the problem as soon as I got out of bed*

The highlighted verbs assumed a figurative meaning when used in contexts related to **emotion** (such as *lock* and *steamroller*, respectively in a positive and a negative way) or *economics* (such as *climb*). This confirms patterns that we have observed several times so far. Besides, the physical verb *attack* has been recognized as metaphorical in two cases that represent figures of speech become very common nowadays. The most frequently recurring nouns are instead almost the same as in the previous analyzed cases, and they are related to the aforementioned topics/contexts:

1. **people** - 4 occurrences;
2. **idea** - 3 occurrences;
3. **problem** - 3 occurrences;
4. **word** - 3 occurrences;

5. **market** - 3 occurrences.

Wrong Metaphorical Predictions This time, among the MOH-X mistakenly classified metaphors there are no words related to the typical topics of correctly classified ones. In some cases, there is a fine line between figurative and literal meaning in the original annotation which might have been the cause of the wrong prediction, such as in the following sentence:

- She wanted to **buy** his love with her dedication to him and his work - *buy his love* here could in fact be considered a metaphor, but the verb was labeled as literal.

In other instances, the words that were mistakenly labeled as metaphors by the model are not so common verbs:

1. *I **inherited** my good eyesight from my mother*;
2. *We **injected** the glucose into the patient's vein.*

VUA Predictions

Correct Metaphorical Predictions With this 1990 *CoHa Word SGNS* model slice, **no words** contained in sentences of the *news* genre have been correctly classified as metaphors. The order of the other genres in terms of frequency is the same we have observed in previous analyses (conversation, fiction, academic). All the most frequent pos-seq tags sequences combinations of these predictions present a *[PRON]*, a *[PROPN]* or a *[NOUN]* followed by a *[VERB]*, and two of them are the ones that we have encountered in all previous analyses:

1. ['PRON', 'VERB', 'PUNCT'];
2. ['PROPN', 'VERB', 'PUNCT'].

Wrong Metaphorical Predictions This time, we have a good number of sentences of the *news* genre. Once again, the model managed to recognize the parts of the sentences which contained figurative expressions, but the annotators chose to label just some of those words leading to erroneous predictions. The pos-seq tags sequences' combinations are not distinct enough to justify a specific analysis.

TroFi Predictions

Correct Metaphorical Predictions The verbs that we see in these correctly made predictions are the same as the ones we found with the other previous embeddings (*Full CoHa CADE*, *CoHa Word CADE 1990 slice* and *GloVe*). A lot of these verbs have **physical** connotations (*strike, attack, kill*), and their metaphorical nature is even in this case determined by their use in **economical** and **emotional** contexts. Wrong Metaphorical Predictions Some of the verbs are the same that we found in the correct predictions. Just like in previous analyses, we encounter instances in which metaphorical labels are assigned to verbs used in literal ways, such as in this sentence:

- *At hearing this morning before a Delaware chancery court judge, Doskocil is **expected** to ask the court to **strike** down Wilson's "poison pill" anti - takeover measures, saying an auction for the company has continued for a sufficient time to allow all bidders to emerge* - both **expected** and **strike** have been classified as metaphors, but they were originally labeled as literal words.

Here, sometimes the predictions end up being wrong as whole because even if the reference verbs were correctly classified, a label of 1 was assigned to other literal verbs as well:

1. *Issues of race and racism would seem to be central to the Duke Ellington story, but film maker Carter **touches** on them only enough to **tantalize** us, no more* - **touches** is correctly classified as metaphor, but **tantalize** was a literal verb;

2. *Mr. Norris added that silver's sharp fall can be attributed to a series of technical levels that it **hit**, **touching** off heavy selling by commodity funds* - **touching** is correctly classified as metaphor, but **hit** was a literal verb.

Nearest Neighbors Analysis

Nearest neighbors analysis on *CoHa HistWords/SGNS 1990 Slice* through Word2Vec lead to results and observations very similar to the ones obtained with *CoHa Word CADE 1990 Slice*. This is probably due to the fact that the two slices span the same decade.

Observations

To summarize, for the three datasets:

1. **MOH-X**: Although we observed verbs that we did not find in previous analyses, the patterns related to topics and other observations are similar to the ones observed before.

2. **VUA**: In the **correct predictions**, no sentences belonging to the *news* British National Corpus genre have been found. *PROPN* (proper noun), *PRON* (pronoun), or *NOUN* followed by a *VERB* are always the most frequent combinations of pos-seq tags sequences in the predictions. Even in this case, some equivocal annotations lead to words being erroneously given a metaphorical label.

3. **TroFi**: The verbs that we see in the correct predictions are the same as the ones we found with the other previous embeddings, with similar patterns. Predictions generally end up being wrong as whole because metaphorical labels have been assigned to literal verbs, or because even if the reference verbs were correctly classified, a label of 1 was assigned to other literal verbs as well.

3.5.7 Conclusions

The various qualitative analyses that we simultaneously performed both on the predictions of all three state of the art metaphor datasets and on the embeddings themselves, revealed and confirmed several fascinating patterns. The most important ones are the following:

1. **Topics related to *economics*, *politics* and *emotions* are the most recurring ones in sentences containing correctly identified metaphors. We could also notice this in the fine-tuning experiments with contextual embeddings, even though they were performed using a different test set, so this is a very important confirmation**;

2. **We observed that verbs having a literal meaning characterized by physical connotations, often assume metaphorical/figurative meanings when used in sentences related to the contexts listed before. This pattern was at first observed especially in TroFi predictions, but thanks to the nearest neighbors analyses we were able to detect it even in the other datasets**;

3. **In several cases, words that are similar to verbs or nouns related to one of the aforementioned contexts, are found in sentences alongside correctly classified metaphorical words that are connected to another one of those most recurring topics**;

4. **Different results related to the domains of the sentences have been observed in VUA dataset's predictions. With *Full CoHa CADE* embedding, only one sentence belonging to the *academic* genre was correctly classified, whereas as far as *CoHa SGNS 1990 slice* is concerned, no sentences belonging to the *news* genre were correctly predicted. The latter result could indicate that for that specific time period, SGNS words' representations of the *news* genre are biased towards their metaphorical meaning (words are used in metaphorical contexts way more than in literal ones). This would prevent the RNN-based models from correctly identifying the words as metaphors. In fact, the models generally recognize metaphors because of the mismatch between the words' signals and those of the contexts of the sentences in which they are located**.

Chapter 4

BERT and Fine-tuning Experiments

4.1 Fine-tuned BERT Model for Metaphor Classification

In recent years there have been many breakthoughs in Natural Language Processing, especially those related to **transfer learning**. Architectures such as *ELMo* ([129]) and *Transformers* ([49]) have allowed researchers to achieve state of the art performances on multiple tasks and provided large pre-trained models with high performance to the community. Since BERT representations already take advantage of the Attention mechanism, it is not particularly useful to exploit them inside the recurrent neural networks-based models. Therefore, we decided to use BERT to tackle metaphor detection task from a different angle: instead of identifying metaphorical words inside the sentences, we tried to recognize and distinguish entire metaphorical sentences from literal ones (thus performing metaphor detection as a *binary classification* problem). In particular, the HuggingFace Transformers library ([140]) has been used, and a BERT model has been fine-tuned on the previously used datasets to make predictions on a new one, from which it was possible to gather useful information about the detected sentences. BERT performance was also compared to a baseline model, where a *TF-IDF vectorizer* and a *Naive Bayes* classifier were used for a preliminary assessment of accuracy. The research questions are the following:

1. Can transfer learning applied to a metaphor dataset in order to make predictions on a different one provide good or better-than-state-of-the-art results for metaphor detection? Is it possible to identify particular patterns between the recognized metaphorical sentences and other information related to them, such as sentiment and polarity connotations?

2. What are the characteristics of correctly and mistakenly metaphorical predictions obtained through our temporal metaphor detection experiments? In particular, are there any specific patterns related to emotion, sentiment and/or source and target concepts? For example, are metaphorical sentences classified as positive in nature more easily identified than negative ones, or vice versa? Could it be that metaphorical sentences concerning a particular topic get correctly identified with higher precision compared to others?

The aim of this experiment was to fine-tune the pre-trained BERT model on our MOH-X, TroFi and VUA datasets, used for training and evaluation, and see whether

predictions made on previously unseen metaphorical and literal sentences coming from the LCC English Dataset (used for test) were accurate. In picture 4.1, we can see how a BERT model is typically trained on a specific spam classification task with a labeled dataset; in our case, the final predictions of the BERT classifier are instead *metaphorical* or *literal*.

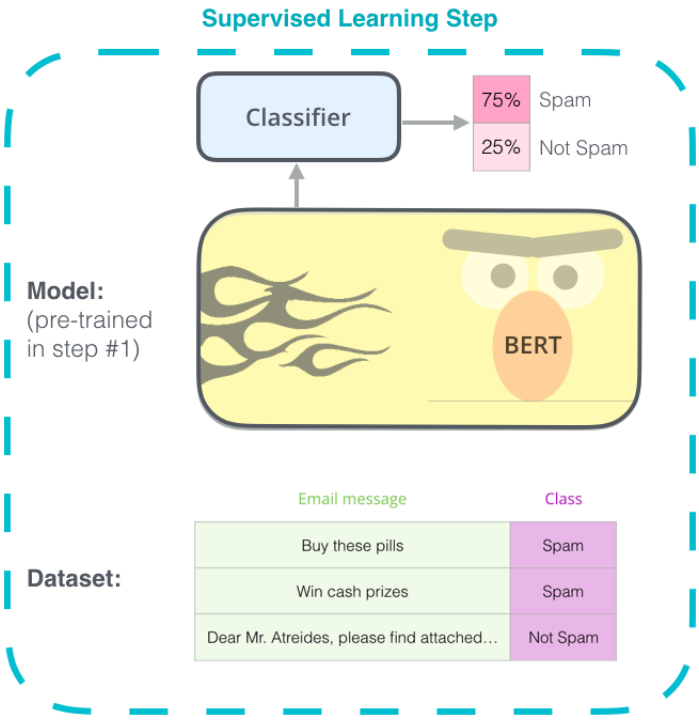

Figure 4.1: Fine-tuning of a BERT model on a labeled dataset for a spam classification task.

The three datasets used for training and evaluation contain sentences that are either metaphorical or literal in nature; the label indicates their nature for the classification. Since only the text data is used for classification, unimportant columns were dropped, with only the sentence and label ones being kept. The test data instead is the LCC English Metaphor Dataset [80]. In order to extract useful information related to the predictions at the end of the experiment, we kept the *polarity*, *intensity*, *source* and *target concepts* columns. The missing values were a very small part of the data (since the LCC datasets were specifically created for research purposes) and therefore they were removed. The LCC test set contains 22094 sentences. 9872 samples are metaphorical, while the remaining 12222 are literal. The training data was first split into two sub-sets: a train set with 90% of the data and a validation set with 10% of the data. The validation set was used to compare the models. The code for this approach has been run on Google Colab, setting a GPU as device. A Naive Bayes model was used as classifier to obtain a baseline performance. For the text preprocessing part, a bag-of-words model was used ([142]): a word is represented as the 'bag' of its words, disregarding grammar and word order. The datasets were already pretty clean, but usual steps related to stop words, punctuations and not useful characters removal were applied anyway. The HuggingFace Transformer library contains a PyTorch ([143]) implementation of state-of-the-art NLP

models including *BERT* ([74]). In order to apply the pre-trained BERT model, we had to use the tokenizer provided by the library. The model has a fixed vocabulary and the BERT tokenizer has a specific way of handling out-of-vocabulary words. We also needed to add special tokens to the start and end of each sentence, pad and truncate all sentences to a single constant length, and explicitly specify what the padding tokens were with a mask which prevents the application of attention mechanism on the padding tokens. The encoding method of BERT tokenizer is used to:

1. Split our text into tokens;
2. Add the special [CLS] and [SEP] tokens;
3. Convert the input sentence in tokens, and the tokens into indexes of the tokenizer vocabulary;
4. Pad or truncate sentences to maximum length;
5. Create the attention mask.

Then, we specified the maximum length of our sentences and tokenized the data. An iterator for the dataset was created to save memory during the training phase and increment its speed at the same time. A this point, the BERT model was defined. The pre-trained *BERT-base* model is made of 12 transformer layers, where each Transformer layer takes in a sequence of token embeddings, producing the same number of embeddings with the same hidden size (or dimensions) on the output. The output of the last Transformer layer of the [CLS] token is used as the features of the sequence to feed a classifier. The *BertForSequenceClassification* class from the Transformers library is specifically designed for classification tasks, but in this case a custom class was defined in order to be able to freely specify the various classifiers. The *BertClassifier* class consists of a BERT model to extract the last hidden layer of the [CLS] token and a single-hidden-layer feed-forward neural network used as the classifier. AdamW optimizer and standard hyper-parameters have been used. It was needed to specify hidden size of BERT, hidden size of the classifier and number of labels. At this point, we trained the model on the entire training data, obtained concatenating train and validation data. In the *training process*, data gets unpacked from the dataloader and loaded onto the GPU. A forward pass is performed to compute logits and loss, then gradients are calculated, and a backward pass is computed to determine these gradients ('exploding gradients' are avoided by clipping the norm of the gradients to 1.0). At this point, both the model's parameters and learning rate are updated. In the *evaluation phase*, the data is again unpacked and loaded onto the GPU; at this point there is a forward pass; then, loss and accuracy rate are computed over the validation set. Finally, we got to the the final part: *evaluation on test set*. Before making predictions on the test set, this data needed to be processed and encoded in the same way as the train data (a custom function was coded exactly for this reason). As far as the *evaluation on test set* is concerned, the prediction step is similar to the one performed in the training loop: a forward pass is in fact utilized to compute logits and a *softmax function* is applied in order to calculate probabilities.

4.1.1 Language Computer Corporation (LCC) Datasets

We managed to collect two of the Language Computer Corporation Metaphor datasets (*Introducing the LCC Metaphor Datasets*, by Michael Mohler et al., [80]), respectively in

English and Spanish languages (the Russian and Farsi ones were not available). We will use the English LCC dataset for our experiments. These datasets were produced over the course of three years by a staff of nine annotators working in the four aforementioned languages, and are characterized by clear sentiment and polarity connotations. The most interesting and useful information provided by these datasets are:

1. **Metaphoricity scores** related to word-pairs contained in the sentences (on a four-point scale ranging from -1.0 to 3, where -1.0 stands for absolutely not metaphorical and 3 stands for highly metaphorical);
2. 114 **source concept** domains and 32 **target concept** domains in total;
3. **Polarity scores** (negative, neutral and positive);
4. **Intensity** (ranging from 0.0 to 3.0).

The metaphoricity scores were transformed according to our classification tasks' needs: negative scores were converted to 0 (non-metaphorical sentence) while positive ones were converted to 1 (metaphorical sentence).

4.1.2 Overall Performances

The BERT classifier has been fine-tuned on MOH-X, VUA and TroFi datasets. Due to time constraints, we explored only the predictions on the LCC test made by the model which was fine-tuned on MOH-X dataset, but we still reported each dataset's performances. The BERT model fine-tuned on MOH-X dataset was able to correctly predict **8515 of the 9872** LCC dataset's metaphorical sentences.

4.1.3 MOH-X - Results and Predictions on LCC Test Set

By combining TF-IDF and the Naive Bayes algorithm, we achieved an accuracy rate of around 71% on the validation set, which represented the baseline performance to be used to evaluate the fine-tune BERT model.

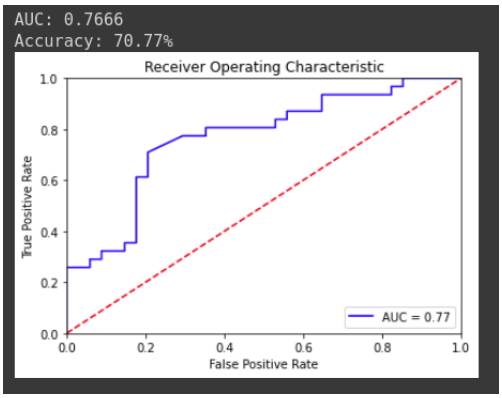

Figure 4.2: MOH-X: accuracy rate of 71% on the validation set obtained through TF-IDF and Naive Bayes algorithm

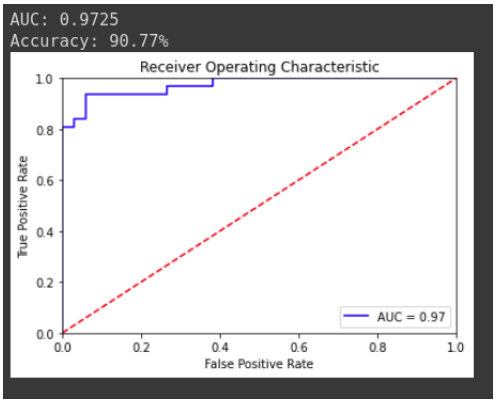

Figure 4.3: MOH-X: accuracy rate of 91% on the validation set obtained through the BERT classifier

The BERT classifier for MOH-X dataset was fine-tuned only for 3 epochs: BERT is already trained with a huge amount of information about the English language. The Bert Classifier achieved 90.77% accuracy rate on the validation set: this score was much higher than the baseline one. In pictures 4.4 and 4.5, we can see the classification report and the confusion matrix for our model:

MOH-X	Precision	Recall	F1-Score	Accuracy
Literal Class	92.00%	97.00%	94.00%	94.00%
Metaphorical Class	97.00%	90.00%	93.00%	94.00%

Figure 4.4: MOH-X: classification report.

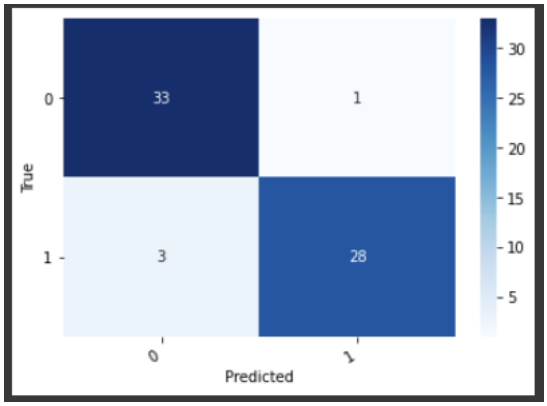

Figure 4.5: MOH-X: confusion matrix.

The results are shown in the classification report (see figure 4.4). As we can see, Transfer learning achieved very good results for metaphor detection task using the MOH-X dataset in a short amount of time and with just a small amount of train set data. Let's

now have a look at the predictions made by the best performing BERT model, which was fine-tuned on the MOH-X dataset over the LCC English Dataset, used as test set. Obtaining the Language Computer Corporation English Metaphor Dataset and being able to use it as our test set for the transfer learning task was very important, since it contains much more information than the standard MOH-X, VUA and TroFi datasets, especially data related to sentiment/polarity, intensity, source and target concepts of the sentences. Therefore, we could gather useful information about both correctly and mistakenly predicted metaphors.

Correctly Classified Metaphors

Let's observe the characteristics of the metaphorical sentences of the LCC English Metaphor dataset that were correctly recognized as such by our fine-tuned BERT model.

1 - Polarity:

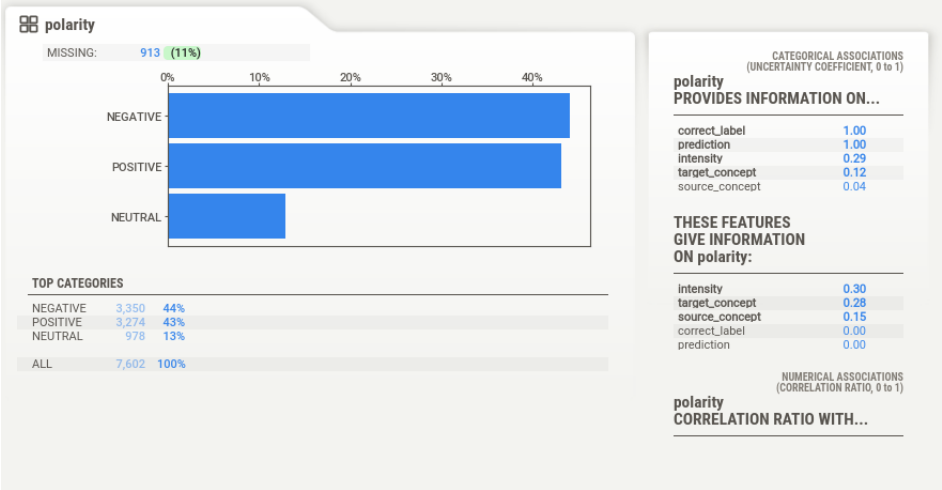

Figure 4.6: LCC: polarity of correctly classified metaphorical sentences.

The metaphorical sentences with a negative connotation were slightly easier for our model to identify than those with a positive one; neutral sentences are the less numerous among the predictions. We already saw that metaphorical/figurative language does have a strong correlation with emotion, and this seems to confirm it: besides, if we observe the features of LCC dataset that are influenced the most by *polarity* (on the right of figure 4.6), we can see that the predicted label is indeed the first one among with the originally correct label of the test set. This is a clear sign that emotion and sentiment have a strong impact on metaphoricity and its detection in text.

2 - Intensity:

Figure 4.7 shows that the sentences characterized by the second highest *intensity* (as stated by the dataset's annotators) are the most numerous among the correctly classified

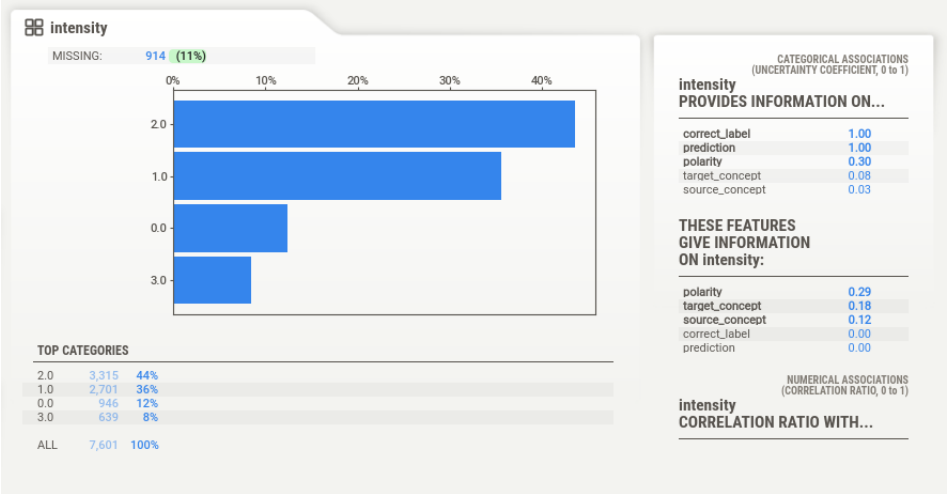

Figure 4.7: LCC: intensity of correctly classified metaphorical sentences.

metaphorical ones. Interestingly enough, the sentences with the highest *intensity* are less numerous than those with null intensity among the correctly classified metaphorical ones. In a similar way compared to *polarity*, even *intensity* strongly impacts the predicted label, among with the original correct one: metaphoricity and sentiment are indeed strictly correlated.

3 - Source and Target Concepts:

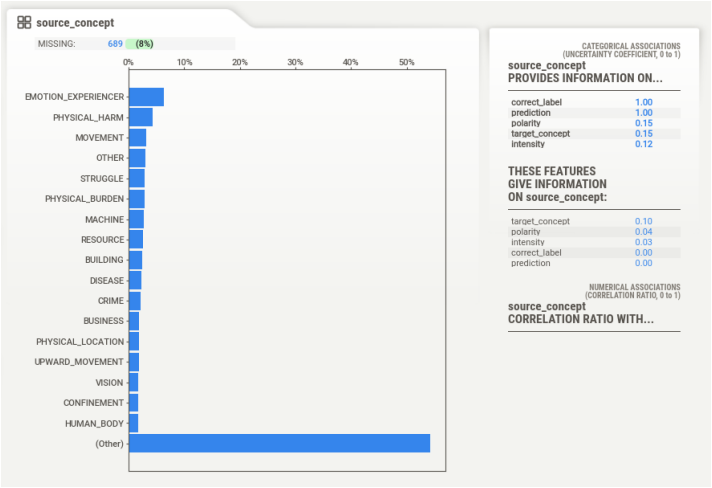

Figure 4.8: LCC: source concept of correctly classified metaphorical sentences.

While the most common *source concepts* among the correctly classified metaphorical sentences are mainly related to emotional and physical notions, their *target concepts* seem to be mostly related to business and politics (see figures 4.8 and 4.9).

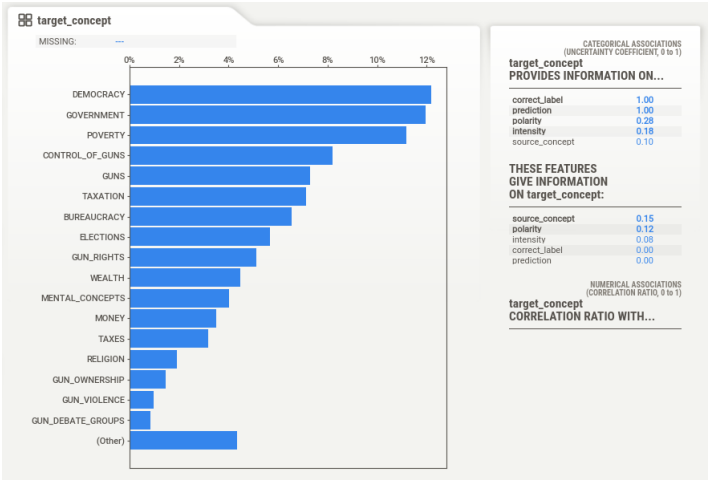

Figure 4.9: LCC: target concept of correctly classified metaphorical sentences.

Mistakenly Classified Metaphors

Now let's observe the characteristics of the metaphorical sentences of the LCC English Metaphor dataset that were mistakenly identified as such by our fine-tuned BERT model.

1 - Polarity:

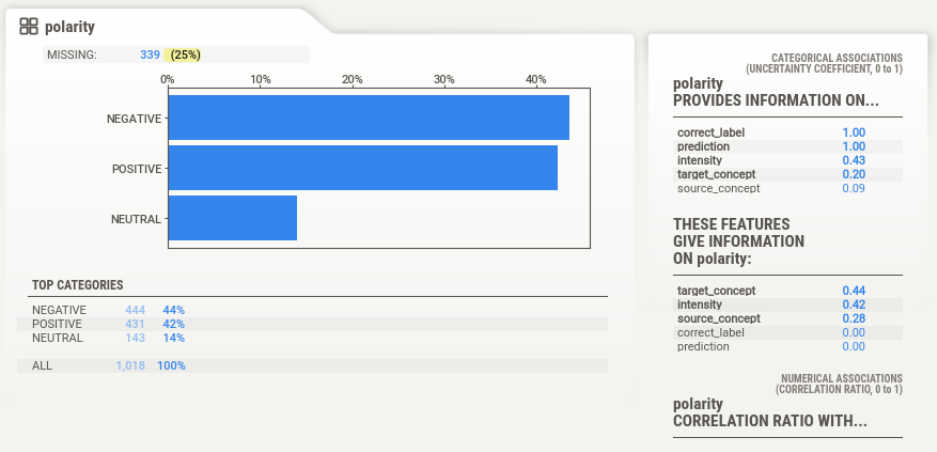

Figure 4.10: LCC: polarity of erroneously classified metaphorical sentences.

The conclusions related to *polarity* are the same as the ones that we drew before (see figure 4.10).

2 - Intensity:

Even for information related to sentences' *intensity*, the observations are the same as before (see figure 4.11).

3 - Source and Target Concepts:

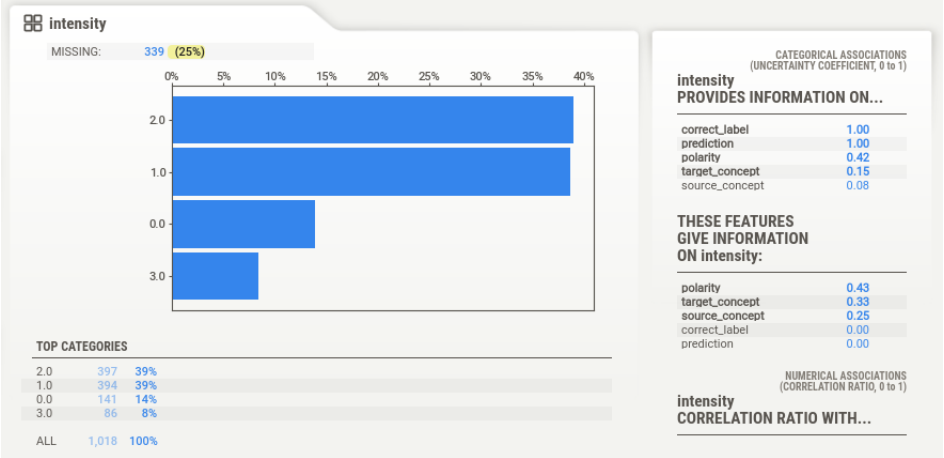

Figure 4.11: LCC: intensity of erroneously classified metaphorical sentences.

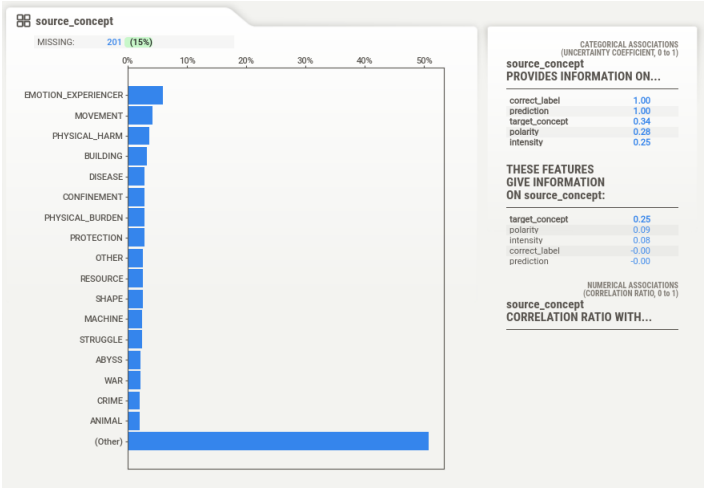

Figure 4.12: LCC: source concept of erroneously classified metaphorical sentences.

While the most common *source concepts* among the mistakenly classified metaphorical sentences are always mainly related to emotional and physical notions, this time their *target concepts* seem to be mostly related to war, politics, the mind and philosophy (see figures 4.12 and 4.13).

4.1.4 VUA - Results

By combining TF-IDF and the Naive Bayes algorithm, we achieved an accuracy rate of around 72% on the validation set, which represented the baseline performance to be used to evaluate the fine-tune BERT model. The BERT classifier was fine-tuned only for 3 epochs. The Bert Classifier achieved 74.74% accuracy rate on the validation set: this score was only a small improvement over the baseline Naive Bayes one. As anticipated before, due to time constraints, we did not proceed further by analyzing the predictions made by the BERT model fine-tuned on this dataset on the LCC test set.

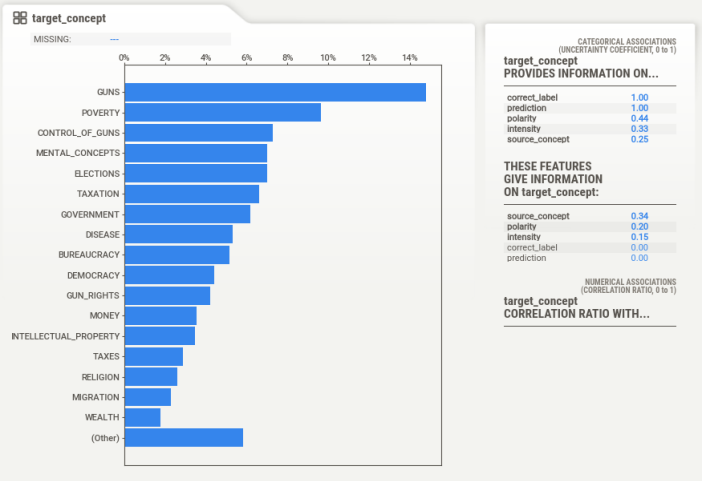

Figure 4.13: LCC: target concept of erroneously classified metaphorical sentences.

4.1.5 TroFi - Results

By combining TF-IDF and the Naive Bayes algorithm, we achieved an accuracy rate of around 58% on the validation set, which represented the baseline performance to be used to evaluate the fine-tune BERT model. The BERT classifier was fine-tuned only for 3 epochs. The Bert Classifier achieved 72.99% accuracy rate on the validation set: this score was much higher than the baseline Naive Bayes. As anticipated before, due to time constraints, we did not proceed further by analyzing the predictions made by the BERT model fine-tuned on this dataset on the LCC test set.

4.2 Conclusions

We collected new data (the Language Computer Corporation dataset) containing both metaphorical sentences and sentiment information, to investigate any new possible patterns highlighted by metaphor detection approaches' predictions. In order to do so, we used contextual word embeddings, specifically BERT, exploiting them within a fine-tuned model trained on the three state of the art datasets. The goal was to see whether the models could correctly classify whole metaphorical sentences in the newly acquired test set, while gathering useful information about them, such as links to feelings and emotion, source and target concepts, and so on. Therefore, this time we approached metaphor detection as a binary classification task. Due to time constraints we were not able to analyze the predictions made on the test set by the fine-tuned model trained on all three state of the art datasets, but the obtained results confirmed the context-related patterns discovered in previous analyses conducted with the other types of embeddings. Emotional and socio-economics topics are the most recurring ones in correctly classified metaphorical sentences.

Chapter 5

Conclusions and Future Research

In this dissertation, we investigated the world of figurative language and metaphors in natural language processing (NLP).

We performed metaphor detection as a sequence classification task in order to identify words with figurative meanings inside sentences, and as a binary classification task to distinguish metaphorical sentences from literal ones. The sequence classification approach, which is the one we experimented with the most, allowed us to take advantage of different types of word representations, especially temporal ones, and evaluate their impact on metaphor detection. Looking at the numerous and diversified results, we can affirm that temporal embeddings do generally improve metaphor detection's performances, although their overall impact on the task is rather limited.

Finally, we performed qualitative analyses on the predictions made by sequence classification neural networks-based approaches and fine-tuned models for metaphor detection. The results suggest that topics related to economics, politics and emotions are the most recurring ones in sentences containing correctly identified metaphors, and that verbs having a literal meaning characterized by physical connotations, often assume a metaphorical meaning when used in sentences related to the aforementioned contexts. We also observed that words' representations of some language domains in specific time periods could be biased towards their metaphorical meaning, leading to words being used in metaphorical contexts way more than in literal ones. This would prevent the neural networks from correctly identifying the words as metaphors, since these models generally recognize metaphors because of the mismatch between the words' signals and those of the contexts of the sentences in which they are located.

Therefore, future research could start from the creation of a new ad hoc dataset based on words with known semantic changes over time. By doing so we would avoid using too many metaphors that have become common figures of speech by now, and we could confirm or deny our hypotheses by checking the new predictions made by the neural networks. Finally, further experimentation with BERT could be performed, since contextual representations are becoming more and more popular and effective in a myriad of current NLP applications.

Bibliography

[1] René Magritte. (1929). *Les mots et les images: choix d'écrits.*

[2] Ludwig Wittgenstein. (1953). *Philosophical Investigations.*

[3] https://www.bbc.com/news/world-us-canada-12385455

[4] William Shakespeare. (1597). *Romeo and Juliet*, Act 2 Scene 2.

[5] George Lakoff and Mark Johnson. (1980). *Metaphors we live by.*

[6] Dan Fass. (1997). *Processing Metonymy and Metaphor (Contemporary Studies in Cognitive Science & Technology).*

[7] Ekaterina Shutova. (December 2015). *Design and Evaluation of Metaphor Processing Systems.* In *Computational Linguistics, Volume 41, Issue.*

[8] Yorick Wilks. (December 1978). *Making Preferences more active.* In *Artificial Intelligence Volume 11, Issue 3, Pages 197-223.*

[9] Wim Peters Ivonne Peters, Piek Vossen. (1998). *Automatic Sense Clustering in EuroWordNet.*

[10] Pragglejaz Group. (1997). *MIP: A Method for Identifying Metaphorically Used Words in Discourse.*

[11] Turney et al. (2010). *From Frequency to Meaning: Vector Space Models of Semantics.*

[12] Del Tredici, Nissim and Zaninello. (2016). *Tracing metaphors in time through self-distance in vector spaces.*

[13] Ekaterina Shutova, Lin Sun. (June 2013). *Unsupervised Metaphor Identification Using Hierarchical Graph Factorization Clustering.* In *Association for Computational Linguistics, Atlanta, Georgia, pages 978-988.*

[14] Ilana Heintz, Ryan Gabbard, Mahesh Srivastava, Dave Barner, Donald Black, Majorie Friedman, Ralph Weischedel. (June 2013). *Automatic Extraction of Linguistic Metaphors with LDA Topic Modeling.* In *Association for Computational Linguistics, Atlanta, Georgia, pages 58-66.*

[15] Dirk Hovy, Shashank Srivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huying Li, Whitney Sanders, Eduard Hovy. (June 2013). *Identifying Metaphorical Word Use with Tree Kernels.* In *Association for Computational Linguistics, Atlanta, Georgia, pages 52-57.*

[16] Beckner, Clay Ellis, Nick C. Blythe, Richard Holland, John Bybee, Joan Ke, Jinyun Christiansen, Morten H. Larsen-Freeman, Diane Croft, William Schoenemann, Tom. (2009). *Language is a complex adaptive system: Position paper.*. In *Language Learning, 59(Suppl 1), 1–26.*.

[17] Francesco La Mantia, Ignazio Licata, Pietro Perconti. (2017). *Language in Complexity.*

[18] Albert Bastardas-Boada Àngels Massip-Bonet. (January 2013). *Complexity perspectives on language, communication and society.*

[19] Dirk Geeraerts. (1997). *Diachronic prototype semantics. A contribution to historical lexicology.*

[20] Dirk Geeraerts, Stefan Grondelaers Dirk Speelman. (1999). *Convergentie en divergentie in de Nederlandse woordenschat. Een onderzoek naar kleding- en voetbaltermen.*

[21] Elizabeth Closs Traugott, Richard B. Dasher. (2001). *Regularity in Semantic Change.*

[22] Bernd Heine, Ulrike Claudi Friederike Hünnemeyer. (1991). *Grammaticalization: a conceptual framework.* In *Chicago: University of Chicago Press, Pp. x+318.*

[23] Jatowt et al. (2014). *A framework for analyzing semantic change of words across time.* In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries, Pages 229–238.*

[24] Mitra et al. (April 2015). *An automatic approach to identify word sense changes in text media across timescales.* In *Natural Language Engineering -1(5):1-26.*

[25] William L. Hamilton, Jure Leskovec, Dan Jurafsky. (2016). *Cultural Shift or Linguistic Drift? Comparing Two Computational Measures of Semantic Change.* In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 2116–2121, Austin, Texas.*

[26] Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, Hui Xiong. (2017). *Dynamic Word Embeddings for Evolving Semantic Discovery.* In *the International Conference on Web Search and Data Mining (WSDM 2018).*

[27] John Sinclair. (2005). *Corpus and text – basic principles.*

[28] Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, Slav Petrov. (July 2012). *Syntactic Annotations for the Google Books NGram Corpus.* In *Proceedings of the ACL 2012 System Demonstrations, pages 169-174).*

[29] Peter Turney, Yair Neuman, Dan Assaf, Yohai Cohen. (July 2011). *Literal and Metaphorical Sense Identification through Concrete and Abstract Context.* In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Pages 680-690).*

[30] Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, Chris Dyer. (June 2014). *Metaphor Detection with Cross-Lingual Model Transfer.* In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Pages 248-258.*

[31] Beata Beigman Klebanov , Chee Wee Leong , Michael Heilman , Michael Flor. (2014). *Different Texts, Same Metaphors: Unigrams and Beyond.*

[32] Hyeju Jang, Yohan Jo, Qinlan Shen, Michael Miller, Seungwhan Moon, Carolyn Rosé. (August 2016). *Metaphor Detection with Topic Transition, Emotion and Cognition in Context.* In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Pages 216-225.*

[33] Rui Mao, Chenghua Lin, Frank Guerin. (2018). *Word Embedding and WordNet Based Metaphor Identification and Interpretation.* In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) Pages 1222-1231.*

[34] Dan Fass. (1991). *met*: A Method for Discriminating Metonymy and Metaphor by Computer.* In *Computational Linguistics, Volume 17, Number 1, March 1991, Pages 49-90.*

[35] Julia Birke, Anoop Sarkar. (April 2006). *A Clustering Approach for Nearly Unsupervised Recognition of Nonliteral Language.* In *11th Conference of the European Chapter of the Association for Computational Linguistics.*

[36] Ekaterina Shutova, Lin Sun. (2010). *Metaphor Identification Using Verb and Noun Clustering.* In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China.*

[37] Saisuresh Krishnakumaran, Xiaojin Zhu. (2007). *Hunting Elusive Metaphors Using Lexical Resources..* In *Proceedings of the Workshop on Computational Approaches to Figurative Language, Pages 13-20.*

[38] Zachary J. Mason. (March 2004). *CorMet: A Computational, Corpus-Based Conventional Metaphor Extraction System.*

[39] Beata Beigman Klebanov, Michael Flor. (2013). *Word association profiles and their use for automated scoring of essays.* In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Pages 1148-1158.*

[40] Bogdan-Ionut Cirstea; Costin-Gabriel Chiru. (2013). *Metaphor Detection.* In *2013 19th International Conference on Control Systems and Computer Science.*

[41] Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. (2018). *Neural Metaphor Detection in Context.* In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Pages 607-613).*

[42] Sepp Hochreiter and Jürgen Schmidhuber. (1997). *Long Short-term Memory.*

[43] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. (2018). *Deep contextualized word representations.*

[44] Alex Graves, Jurgen Schmidhuber. (2005). *Framewise Phoneme Classification with Bidirectional LSTM Networks.*

[45] Jeffrey Pennington, Richard Socher, Christopher D. Manning. (2014). *GloVe: Global Vectors for Word Representation.* In *Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543. URL - http://www.aclweb.org/anthology/D14-1162.*

[46] M. Schuster and K. K. Paliwal. (1997). *Bidirectional recurrent neural networks.*

[47] P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. (1999). *Exploiting the past and the future in protein secondary structure prediction.*

[48] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. (2015). *Neural Machine Translation by Jointly Learning to Align and Translate.* In *Computation and Language (cs.CL); Machine Learning (cs.LG); Neural and Evolutionary Computing (cs.NE); Machine Learning (stat.ML).*

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. (2017). *Attention Is All You Need.*

[50] Saif M. Mohammad, Ekaterina Shutova, Peter D. Turney. (2016). *Metaphor as a Medium for Emotion: An Empirical Study.*

[51] Ekaterina Shutova, Douwe Kiela and Jean Maillard. (2016). *Black Holes and White Rabbits: Metaphor Identification with Visual Features.* In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Pages 160-170.*

[52] Christiane Fellbaum. (1998). *A Semantic Network of English: The Mother of All WordNets.*

[53] Chee Wee (Ben) Leong, Beata Beigman Klebanov, Ekaterina Shutova. (June 2018). *A Report on the 2018 VUA Metaphor Detection Shared Task.* In *Proceedings of the Workshop on Figurative Language Processing, Pages 56-66.*

[54] Gerard Steen, Lettie Dorst, Anna Kaal, J. Berenike Herrmann. (2010). *A method for lin- guistic metaphor identification: From MIP to MIPVU.*

[55] Yael Karov, Shimon Edelman. (1998). *Similarity-based Word Sense Disambiguation.* In *Computational Linguistics, Volume 24, Number 1, Special Issue on Word Sense Disambiguation, Pages 41-59.*

[56] Adwait Ratnaparkhi. (1996). *A Maximum Entropy Model for Part-Of-Speech Tagging.* In *Conference on Empirical Methods in Natural Language Processing.*

[57] Aravind K. Joshi. (1999). *Supertagging: An Approach to Almost Parsing.*

[58] William L. Hamilton, Jure Leskovec, Dan Jurafsky. (2016). *Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change*

[59] https://www.english-corpora.org/

[60] Omer Levy, Yoav Goldberg, Ido Dagan. (2015). *Improving Distributional Similarity with Lessons Learned from Word Embeddings.*

[61] Zellig S. Harris. (1954). *Distributional Structure.*

[62] J. R. Firth. (1957). *Applications of General Linguistics.*

[63] Peter D. Turney, Patrick Pantel. (2010). *From Frequency to Meaning: Vector Space Models of Semantics.*

[64] Lin et al. (2012). *Choosing Transfer Languages for Cross-Lingual Learning.*

[65] Federico Bianchi. (2019) *Corpus-based Comparison of Distributional Models of Language and Knowledge Graphs.*

[66] https://www.corpusdata.org/formats.asp

[67] https://www.english-corpora.org/googlebooks/compare-googleBooks.asp

[68] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, Saif Mohammad. (2014). *NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews.* In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Pages 437-442.*

[69] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng and Christopher Potts. (2013). *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.* In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642.*

[70] Saif M. Mohammad, Svetlana Kiritchenko and Xiaodan Zhu. (August 2014). *Sentiment Analysis of Short Informal Text.*

[71] William M. Pottenger, Lars E. Holzman. (January 2003). *Classification of Emotions in Internet Chat: An Application of Machine Learning Using Speech Phonemes.*

[72] Agosta, Salvatore J., Brooks, Daniel R. (2020). *The Major Metaphors of Evolution.*

[73] Saif M. Mohammad. (2012). *Emotional tweets.* In *SemEval '12: Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, Pages 246-255.*

[74] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*

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

[76] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, Veselin Stoyanov. (2019). *Unsupervised Cross-lingual Representation Learning at Scale.*

[77] Guillaume Lample, Alexis Conneau. (2019). *Cross-lingual Language Model Pretraining.*

[78] Rico Sennrich, Barry Haddow, Alexandra Birch. (2015). *Neural Machine Translation of Rare Words with Subword Units.*

[79] Taku Kudo, John Richardson. (2018). *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing.*

[80] Michael Mohler, Mary Brunson, Bryan Rink, Marc Tomlinson. (2016). *Introducing the LCC Metaphor Datasets.* In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16).*

[81] http://saifmohammad.com/WebPages/metaphor.html

[82] https://arxiv.org/pdf/1906.01502.pdf

[83] https://arxiv.org/pdf/1911.02116.pdf

[84] https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[85] https://stackoverflow.com/questions/48001598/why-do-we-need-to-call-zero-grad-in-pytorch

[86] Yoav Goldberg. (2017). *Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies), page 135.*

[87] Warren Sturgis McCulloch and Warwick Pitts. (1943). *A logical calculus of the ideas immanent in nervous activity. 1943. Bulletin of mathematical biology, 52 1-2:99–115; discussion 73–97.*

[88] Richard H R Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Se- bastian Seung. (2000). *Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000. ISSN 1476-4687. doi: 10.1038/35016072. URL https://doi.org/10.1038/35016072.*

[89] Jeffrey L. Elman. Finding structure in time. (1990). *Cognitive Science, 14(2):179–211.*

[90] Sepp Hochreiter and Jürgen Schmidhuber. (November 1997). *Long short-term memory. Neural Comput., 9 (8):1735–1780. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.*

[91] Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. (2000). *Learning to forget: Continual prediction with lstm. Neural Computation, 12:2451–2471.*

[92] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. (2005). *Neural Networks, 18(5):602 – 610. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2005.06.042. URL http: //www.sciencedirect.com/science/article/pii/S0893608005001206. IJCNN.*

[93] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. (2013). *Efficient estimation of word representations in vector space. In 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, Workshop Track Proceedings. URL - http://arxiv.org/abs/1301.3781.*

[94] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. (2013). *Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc. - URL http://papers.nips.cc/paper/ 5021-distributed-representations-of-words-and-phrases-and-their-compositionalit pdf.*

[95] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. (March 2003) *A neural probabilistic language model. J. Mach. Learn. Res., 3:1137–1155, ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944966.*

[96] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. (2011). *Natural language processing (almost) from scratch. CoRR, abs/1103.0398, 2011. URL http://arxiv.org/abs/1103.0398.*

[97] Joseph Turian, Lev Ratinov, and Yoshua Bengio. (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, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm? id=1858681.1858721.*

[98] Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. (October 2013). *Bilingual word embeddings for phrase-based machine translation. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1393–1398, Seattle, Washington, USA. Association for Computational Linguistics. URL https://www.aclweb. org/anthology/D13-1141.*

[99] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. (2016). *Bag of tricks for efficient text classification. CoRR, abs/1607.01759. URL http://arxiv.org/abs/1607.01759.*

[100] Robert Speer, Joshua Chin, and Catherine Havasi. (2016). *Conceptnet 5.5: An open multilingual graph of general knowledge. CoRR, abs/1612.03975. URL http://arxiv.org/abs/1612.03975.*

[101] https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html

[102] Leonard Bloomfield. (1933). *Language.* Allen Unwin.

[103] Michel Bréal. (1899). *Essai de sémantique.* Hachette, Paris.

[104] Gustaf Stern. (1931). *Meaning and change of meaning; with special reference to the English language.* Wettergren Kerbers.

[105] Andreas Blank and Peter Koch. (1999). *Historical semantics and cognition.* Walter de Gruyter.

[106] Dirk Geeraerts. (1997). *Diachronic prototype semantics: A contribution to historical lexicology.* Clarendon Press, Oxford.

[107] Elizabeth Closs Traugott and Richard B Dasher. (2001). *Regularity in semantic change.* Cambridge University Press.

[108] Joachim Grzega and Marion Schoener. (2007). *English and General Historical Lexicology.* Eichstätt-Ingolstadt: Katholische Universität.

[109] Elizabeth Traugott. (2017). *Semantic change.* Oxford Research Encyclopedias: Linguistics.

[110] M. Hilpert. (2008). *Germanic future constructions: A usage-based approach to language change.* Benjamins, Amsterdam, Netherlands.

[111] Stefan Th. Gries. (1999). *Particle movement: a cognitive and functional approach.* Cognitive Linguis- tics, 10:105–145.

[112] D. Kerremans, S. Stegmayr, and H.-J. Schmid. (2010). *The NeoCrawler: Identifying and retrieving neologisms from the internet and monitoring ongoing change.* In *K. Allan and J. A. Robinson, editors, Current methods in historical semantics, pages 130–160.* De Gruyter Mouton.

[113] John Firth. (1957). *A synopsis of linguistic theory, 1930-1955.* Blackwell.

[114] Kristina Gulordava and Marco Baroni. (2011). *A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus.* In *Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics, pages 67–71, Edinburgh, UK.*

[115] Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. (2015). *Statistically significant detection of linguistic change.* In *Proceedings of the 24th International Conference on World Wide Web, pages 625–635, Florence, Italy.*

[116] Wayne A Taylor. (2000). *Change-point analysis: a powerful new tool for detecting changes.*

[117] Yating Zhang, Adam Jatowt, Sourav Bhowmick, and Katsumi Tanaka. (2015). *Omnia mutantur, nihil interit: Connecting past with present by finding corresponding terms across time.* In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 645–655, Beijing, China.*

[118] Yating Zhang, Adam Jatowt, Sourav S. Bhowmick, and Katsumi Tanaka. (2016). *The past is not a foreign country: Detecting semantically similar terms across time. IEEE Transactions on Knowledge and Data Engineering, 28(10):2793–2807, October.*

[119] Steffen Eger and Alexander Mehler. (2016). *On the linearity of semantic change: Investigating meaning variation via dynamic graph models.* In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 52–58, Berlin, Germany.*

[120] Edouard Grave, Armand Joulin, Quentin Berthet. (May 2018). *Unsupervised Alignment of Embeddings with Wasserstein Procrustes.*

[121] Goodall, C. (1991). *Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society. Series B (Methodological), pages 285–339.*

[122] Mikolov, T., Le, Q. V., and Sutskever, I. (2013). *Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168.*

[123] Xing, C., Wang, D., Liu, C., and Lin, Y. (2015). *Normalized word embedding and orthogonal transform for bilingual word translation.* In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1006–1011.*

[124] Schönemann, P. H. (1966). *A generalized solution of the orthogonal procrustes problem. Psychometrika, 31(1):1–10.*

[125] https://catalog.ldc.upenn.edu/LDC2000T43

[126] http://www.lang.osaka-u.ac.jp/ sugimoto/MasterMetaphorList/metaphors/

[127] https://en.wikipedia.org/wiki/One-hot

[128] https://builtin.com/data-science/step-step-explanation-principal-component-analysis

[129] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. (March 2018). *Deep contextualized word representations.*

[130] Ian Goodfellow, Yoshua Bengio, Aaron Courville. (2016). *Deep Learning (Adaptive Computation and Machine Learning series), page 526.*

[131] https://www.thoughtco.com/hypernym-words-term-1690943

[132] https://plato.stanford.edu/entries/bayes-theorem/

[133] Boleda, G. (2020). *Distributional Semantics and Linguistic Theory. Annu. Rev. Linguist. 6:213–34. DOI: 10.1146/annurev-linguistics-011619-030303.*

[134] https://www.english-corpora.org/coha/

[135] https://en.wikipedia.org/wiki/SQL

[136] https://nlp.stanford.edu/projects/histwords/

[137] Dan Fass and Yorick Wilks. (1975). *Preference Semantics, III-Formedness, and Metaphor.*

[138] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. (2013). *Distributed Representations of Words and Phrases and their Compositionality.*

[139] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, Ivan Titov. (2019). *Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned.*

[140] https://huggingface.co/transformers/

[141] https://towardsdatascience.com/cross-validation-and-hyperparameter-tuning-how-to-optimise-your-machine-learning-model-13f005af9d7d

[142] https://en.wikipedia.org/wiki/Bag-of-words_model

[143] https://pytorch.org/

[144] Mitchell, Tom (1997). *Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.*

[145] The definition 'without being explicitly programmed' is often attributed to Arthur Samuel, who coined the term 'machine learning' in 1959, but the phrase is not found verbatim in this publication, and may be a paraphrase that appeared later. Confer 'Paraphrasing Arthur Samuel (1959), the question is: How can computers learn to solve problems without being explicitly programmed?' in Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A. (1996). *Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design '96.* Springer, Dordrecht. pp. 151–170. doi:10.1007/978-94-009-0279-4_9.

[146] Rumelhart, D.E; McClelland, James (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge: MIT Press. ISBN 978-0-262-63110-5.

[147] https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8_401

[148] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, Thomas Wolf. (June 2019). *Transfer Learning in Natural Language Processing.* In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials.*

[149] https://medium.com/@ncjatin/advantages-and-disadvantages-of-decision-tree-274f32bc8274

[150] https://scikit-learn.org/stable/modules/naive_bayes.html

[151] https://machinelearningmastery.com/better-naive-bayes/

[152] https://towardsdatascience.com/svm-and-kernel-svm-fed02bef1200

[153] https://en.wikipedia.org/wiki/Multi-label_classification

[154] Wang, Y., Hou, Y., Che, W. et al. (2020). *From static to dynamic word representations: a survey. Int. J. Mach. Learn. Cyber. 11, 1611–1630. https://doi.org/10.1007/s13042-020-01069-8.*

[155] Devlin J, Chang MW, Lee K, Toutanova K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.*

[156] McCann B, Bradbury J, Xiong C, Socher R. (2017). *Learned in translation: contextualized word vectors.* In: *Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural Information processing systems 30. Curran Associates, Inc., pp 6294–6305.* http://papers.nips.cc/paper/7209-learned-in-translation-contextualized-word-vectors.pdf

[157] Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. (2018). *Deep contextualized word representations.* In: *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: human language technologies, vol 1 (Long Papers). Association for Computational Linguistics, New Orleans, Louisiana, pp 2227–2237. https://doi.org/10.18653/v1/N18-1202.* https://www.aclweb.org/anthology/N18-1202

[158] Radford A, Narasimhan K, Salimans T, Sutskever I. (2018). *Improving language understanding by generative pre-training.* https://s3-us-west-2amazonaws.com/openai-assets/research-covers/languageunsupervised/language/understanding/paper/pdf

[159] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, Erik Velldal. (June 2018). *Diachronic word embeddings and semantic shifts: a survey.*

[160] Qi Liu, Matt J. Kusner, Phil Blunsom. (March 2020). *A Survey on Contextual Embeddings.*

[161] Bengio et al. (2003). *A neural probabilistic language model.* In *Journal of machine learning research 3 (Feb), pp. 1137–1155.*

[162] Dai and Le. (2015). *Semi-supervised sequence learning.* In *Advances in neural information processing systems, pp. 3079–3087.*

[163] Ramachandran et al.. (2016). *Unsupervised pretraining for sequence to sequence learning. arXiv preprint arXiv:1611.02683.*

[164] Lan et al. (2019). *ALBERT: a lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.*

[165] Rebuffi et al. (2017). *Learning multiple visual domains with residual adapters.* In *Advances in Neural Information Processing Systems, pp. 506–516.*

'Somehow I manage.' - Michael Scott, 'The Office'