UNISINOS

Programa de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Bruna Koch Schmitt

Exploring Linguistic Information and Semantic Contextual Models for a Relation Extraction Task Using Deep Learning

São Leopoldo, 2020

Bruna Koch Schmitt

EXPLORING LINGUISTIC INFORMATION AND SEMANTIC CONTEXTUAL MODELS FOR A RELATION EXTRACTION TASK USING DEEP LEARNING

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Advisor:
Prof. Dr. Sandro José Rigo

São Leopoldo
2020

S355e Schmitt, Bruna Koch.
Exploring linguistic information and semantic contextual models for a relation extraction task using deep learning / por Bruna Koch Schmitt. – 2020.
55 f. : il. ; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2020.
"Advisor: Dr. Sandro José Rigo".

1. Natural language processing. 2. Relation extraction. 3. Deep learning. I. Título.

CDU: 004.8:800.8

Catalogação na Publicação (CIP):
Bibliotecário Alessandro Dietrich - CRB 10/2338

ABSTRACT

Deep Learning (DL) methods have been extensively used in many Natural Language Processing (NLP) tasks, including in semantic relation extraction. However, the performance of these methods is dependent on the type and quality of information being used as features. In NLP, linguistic information is being increasingly used to improve the performance of DL algorithms, such as pre-trained word embeddings, part-of-speech (POS) tags, synonyms, etc, and the use of linguistic information is now present in several state-of-the-art algorithms in relation extraction. However, no effort has been made to understand exactly the impact that linguistic information from different levels of abstraction (morphological, syntactic, semantic) has in these algorithms in a semantic relation extraction task, which we believe may bring insights in the way deep learning algorithms generalize language constructs when compared to the way humans process language. To do this, we have performed several experiments using a recurrent neural network (RNN) and analyzed how the linguistic information (part-of-speech tags, dependency tags, hypernyms, frames, verb classes) and different word embeddings (tokenizer, word2vec, GloVe, and BERT) impact on the model performance. From our results, we were able to see that different word embeddings techniques did not present significant difference on the performance. Considering the linguistic information, the hypernyms did improve the model performance, however the improvement was small, therefore it may not be cost effective to use a semantic resource to achieve this degree of improvement. Overall, our model performed significantly well compared to the existing models from the literature, given the simplicity of the deep learning architecture used, and for some experiments our model outperformed several models presented in the literature. We conclude that with this analysis we were able to reach a better understanding of whether deep learning algorithms require linguistic information across distinct levels of abstraction to achieve human-like performance in a semantic task.

Keywords: Natural Language Processing. Relation Extraction. Deep Learning.

RESUMO

Métodos de Aprendizado Profundo (AP) tem sido usados em muitas tarefas de Processamento de Linguagem Natural (PLN), inclusive em tarefas de extração de relações semânticas. Entretanto, a performance dos métodos é dependente do tipo e qualidade da informação dada ao algoritmo como características. Em PLN, informações linguísticas tem sido cada vez mais usadas para melhorar a performance de algoritmos de AP, como por exemplo, vetores de palavras pré-treinados, marcadores sintáticos, sinônimos, etc, e atualmente o uso de informações linguísticas está presente nos algoritmos de extração de relações do estado da arte. Porém, não tem sido o foco dessas pesquisas entender exatamente o impacto que o uso de informações linguísticas advindas de níveis distintos de abstração (morfológico, sintático, semântico) tem nos algoritmos aplicados a extração de relações, o que em nossa opinião pode trazer um maior conhecimento da forma que algoritmos de aprendizado profundo generalizam construtos da linguagem quando comparados com a forma que humanos processam a linguagem. Para atingir esse objetivo, realizamos vários experimentos usando uma rede neural recorrente e analizamos qual o impacto que informações linguísticas (categorias gramaticais, categorias sintáticas, hiperônimos, *frames* e classes verbais) e *word embeddings* (tokenizer, word2vec, Glove e BERT) tem na performance do modelo. A partir dos nossos resultados, vimos que os diferentes tipos de *word embeddings* não apresentaram uma diferença significativa na performance. Considerando a informação linguística, o uso de hiperônimos demonstrou uma melhora de performance do modelo, porém considerando que a melhora foi pequena, entendemos que pode não haver um melhor custo-benefício em usar esse recurso semântico para atingir uma melhora pequena de performance. De forma geral, nosso modelo atingiu uma performance boa comparada aos modelos da literatura, especialmente dada a simplicidade da arquitetura de aprendizado profundo usada nos experimentos. E ainda para alguns experimentos, nosso modelo teve a performance melhor que modelos apresentados na literatura. Em conclusão, consideramos que com essa análise obtivemos um melhor entendimento no quesito se os modelos de aprendizado profundo se beneficiam de informação linguística oriunda de distintos níveis de abstração linguística para atingir uma performance próxima à humana em uma tarefa semântica.

Palavras-chave: Processamento de Linguagem Natural. Extração de Relações. Aprendizado Profundo.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Recently, Deep Learning (DL) has been extensively used in the Natural Language Processing (NLP) field in a wide range of tasks, ranging from machine translation, question-answering, sentiment analysis, etc, in what is called a third wave in the NLP development (DENG; LIU, 2018).

In addition, DL has been producing state-of-the-art results for several NLP tasks (GOLDBERG, 2017), revolutionizing speech, video, and image processing. And in the most recent years, NLP research using Deep Learning has become a very successful research area which has been showing significant progress (DENG; LIU, 2018).

Relation extraction (RE) is a very prominent area of Information Extraction (IE) (KUMAR, 2017) which aims to identify and classify concepts and their relations in texts written in natural language, which is especially useful for types of texts that are produced in large quantities, such as scientific papers. The idea behind it is to provide a way to automatically extract meaningful information beyond what standard search engines are capable of doing today, therefore improving drastically the quality and content of the information that can be extracted from such corpora.

Current methods use linguistic information as input for the networks, however it is not explored how the different types of linguistic information contribute to the overall performance of the models.

In this research, we intend to review and perform experiments in a relation extraction task that uses scientific corpora. The experiments use a deep learning method and focus on exploring the linguistic information presented to the algorithm and its impact in the performance of the model in classifying the relations in the texts.

1.1 Objectives and Research Questions

The main objective of this research is to use a deep learning method in a relation extraction task, and analyze the impact that linguistic information (such as morphological, syntactic, and semantic nature) has on performing this task. We use a well-known corpus as dataset: the SemEval 2018, Task 7. This dataset provides abstracts extracted from scientific papers with manually and automatically annotated entities and relation between these entities. Our main research question is:

- What is the impact of different linguistic information (morphological, syntactic, semantic) on the performance of a deep learning model in a relation extraction task?

We have the following specific goals with this research:

1. Identify the most commonly-used Deep Learning and traditional (non-DL) methods that

were used for the mentioned task, especially considering which type of linguistic and non-linguistic information was used for each method;

2. Perform experiments with DL methods using linguistic information (morphosyntactic, semantic, distributinal) and different types of word-embeddings;

3. Compare and evaluate our results with the ones from the literature, focusing especially on the impact that different types of linguistic information had in the accuracy of the relation extraction task.

1.2 Methodology

Initially, a preliminary exploratory study of the literature was conducted to understand the approaches used in Relation Extraction, the most commonly-used algorithms and how the algorithms were evaluated in this specific IE task. This study is presented in chapter 3.

This analysis was essential to delimit the scope of the research and to observe trends and gaps. We complemented this investigation with a specific analysis of the linguistic resources and background used in the state-of-the-art approaches.

Regarding the experiments, this is an experimental research, since we performed experiments using a deep learning algorithm with different setups in terms of linguistic features and analyzed their performance and the impact of each set of features. Regarding the objectives, this is an explanatory research, and to achieve that, we follow WAZLAWICK, 2017 to define the following research stages: literature review and analysis; solution proposal taking into consideration the previous analysis; prototype description and evaluation.

Chapter 4 will describe in depth the corpora used for the experiments, the deep learning architecture, the feature engineering process, and the experimental setups.

1.3 Outline

This thesis is structured as: Chapter 2, which will discuss the theoretical background relevant for this research. This chapter is divided into two major sections, one that will cover the deep learning concepts and the second which will cover the linguistic concepts; Chapter 3, which will review and analyze the current state-of-art literature; Chapter 4, which will discuss the dataset, and the deep learning architecture; Chapter 5 which will present the results of the experiments; and at last the Chapter 6, which will present the final considerations of this research.

2 THEORETICAL BACKGROUND

2.1 Relation Extraction in Natural Language Processing

Relation Extraction is a very important and challenging subtask of the field called Information Extraction (IE), which consists of "extracting structured information, that can be interpreted easily by a machine or a program, from plain unstructured text." (KUMAR, 2017, p. 1). This is of paramount importance given the large amounts of textual data present in the Web today, and the need of extracting meaningful facts from it. Current standard query engines have limited capabilities, which automatic relation extraction aims to improve (GÁBOR et al., 2018). Such improvement could be in the way of "finding all papers that address a problem in a specific way, or discovering the roots of a certain idea" (GÁBOR et al., 2018).

Relation extraction addresses the identification and classification of an entity pair and its relation with each other within a predefined set of relations using a set of documents. For instance, in the SemEval 2018, Task 7 dataset[1], one of the semantic relations is PART-WHOLE and it is defined as:

PART_WHOLE is an asymmetrical relation. It holds between two entities X and Y, where, for example:

- X is a part, a component of Y
- X is found in Y
- Y is built from/composed of X

Therefore, given the sentence below (extracted from the mentioned dataset):
Several extensions of this basic idea are being discussed and/or evaluated: Similar to activities one can define subsets of larger database and detect those automatically which is shown on a large ***database*** *of* ***TV shows****.*

The entities *database* and *TV shows* express a relation between them in a way that TV shows (X) is a component/part of entity database (Y). Therefore, after reading the passage, a reader may identify that this sentence contains a reverse semantic relation of PART_WHOLE. We will discuss the relations of the dataset in the Methods section.

This illustrates how the relation extraction is important in extracting information on a deeper semantic level than standard query engines, which cannot provide information on how the entities in a text are related to each other.

This has several applications, of which a prominent one is the IE from scientific corpora. Scientific corpora is produced in large amounts and are published in the Web in a unstructured textual format. However, "empirical research requires gaining and maintaining an understanding of the body of work in specific area. For example, typical questions researchers face are

[1] https://competitions.codalab.org/competitions/17422

which papers describe which tasks and processes, use which materials and how those relate to one another." (AUGENSTEIN et al., 2017, p. 1) Therefore, researchers and professionals are expected to review and be up-to-date with a large number of publications, which are being continually updated.

Search engines such as Google Scholar, Scopus or Semantic Scholar which mainly constitute efforts to fill in the gap, however, they are limited in their capabilities (AUGENSTEIN et al., 2017), which is where the models we will discuss are focused on: extracting meaningful semantic information about the content of large amounts of textual scientific data in order to provide the user answers to questions such as "which materials are used in specific field?".

2.2 Deep Learning Review

This section will overview the field of Deep Learning and the most frequently used methods accordingly to the literature review.

2.2.1 General description of Deep Learning

Deep learning can be defined as:

> "Deep learning is a branch of machine learning. It is a re-branded name for neural networks—a family of learning techniques that was historically inspired by the way computation works in the brain, and which can be characterized as learning of parameterized differentiable mathematical functions. The name deep-learning stems from the fact that many layers of these differentiable function are often chained together." (GOLDBERG, 2017, p. 26)

In the NLP field, the use of deep learning methods is considered as the major third wave in NLP development, after Rationalism and Empiricism, and the use of Deep Learning has been proven successful in many areas such as speech recognition, computer vision, machine translation, image captioning, visual question answering, speech understanding, etc. (DENG; LIU, 2018)

However, in text-based NLP processing, its effectiveness is less clear, given that "in NLP, stronger theories and structured models on morphology, syntax, and semantics have been advanced to distill the underlying mechanisms of understanding and generation of natural languages, which have not been as easily compatible with neural networks." (DENG; LIU, 2018, p. 10)

Because of this, using deep learning to NLP tasks has become the most active area in NLP and deep learning research (DENG; LIU, 2018). Several methods are used in NLP tasks and we will give an overview of them. We will focus on the methods extracted from the literature review that were used in the SemEval 2018 Task 7 dataset.

2.2.2 Deep Learning Methods

This section will review briefly the most relevant deep learning method used in the papers described in the Literature Review section.

- Long Short-Term Memory (LSTM) and variations

  Long Short-Term Memory (LSTM) is a artificial recurrent neural network architecture proposed by (HOCHREITER; SCHMIDHUBER, 1997, p. 1) that "used an efficient, gradient-based algorithm for an architecture enforcing constant (...) error flow through internal states of special units", which are called "memory cells" (TAI; SOCHER; MANNING, 2015). It was "designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs." (SAK; SENIOR; BEAUFAYS, 2014).

- Recurrent Neural Networks (RNN)

  Recurrent Neural Networks (RNN) are a type of artificial neural network proposed by (ELMAN, 1990) "that has activation feedback which embodies short-term memory. A state layer is updated not only with the external input of the network but also with activation from the previous forward propagation. The feedback is modified by a set of weights as to enable automatic adaptation through learning (e.g. backpropagation)." (BODEN, 2002, p. 7)

- Convolutional Neural Networks (CNN)

  According to (KIM, 2014, p. 1):

> Convolutional neural networks (CNN) are deep learning models that utilize layers with convolving filters that are applied to local features (LeCun et al., 1998). Originally invented for computer vision, CNN models have subsequently been shown to be effective for NLP and have achieved excellent results in semantic parsing (Yih et al., 2014), search query retrieval (Shen et al., 2014), sentence modeling (Kalchbrenner et al., 2014), and other traditional NLP tasks (Collobert et al., 2011).

2.3 Linguistic Knowledge and Natural Language Processing

This section will contextualize the different types of linguistic information that humans process and store in order to be able to communicate. The Section 4.3.2 will explain how these concepts relate to the proposed model.

The human mind processes linguistic information at various levels, which constitutes our knowledge of language. These levels traditionally are divided into the following knowledge domains (MARTIN; JURAFSKY, 2009):

- **Phonetics and Phonology:** linguistic sounds
- **Morphology:** meaningful components of words
- **Syntax:** structural relationships between words
- **Semantics:** meaning
- **Pragmatics:** relationship of meaning to the goals and intents of the speaker
- **Discourse:** relationship of linguistic units larger than a single utterance

Together, all these levels of abstract linguistic information are responsible for enabling us to communicate in natural languages. All these components interact in the mind of the speakers, and many theories and formal models have been created since the dawn of Linguistics as a research field.

In Linguistics, the theories and models can be grouped into three schools of thought.

- **Structuralism/Behaviorism:** based on the works of Saussure (SAUSSURE, 1916), Bloomfield (BLOOMFIELD, 1933), and Skinner (SKINNER, 1957).
- **Generativism:** based on the works of Chomsky (CHOMSKY, 1965).
- **Cognitive-functionalism:** based on the works of Langacker (LANGACKER, 1995). Within the cognitive framework, Fillmore (FILLMORE et al., 1976) developed the Semantic Frames, a very prominent semantic theory.

In parallel, in the field of Computational Linguistics (CL) and Natural Language Processing (NLP), several formal models were being developed to train machines to communicate as humans do, but the views on Language and the theories and models derived from Linguistics have impacted the algorithms developed. We will summarize them below citing the impact from the Linguistics theories when relevant.

The NLP development can be divided into three waves (DENG; LIU, 2018), (MARTIN; JURAFSKY, 2009):

- **Rationalism:** this wave dates back from 1950s, and it is based on:

  > "Postulating that key parts of language are hardwired in the brain at birth as a part of the human genetic inheritance, rationalist approaches endeavored to design hand-crafted rules to incorporate knowledge and reasoning mechanisms into intelligent NLP systems." (DENG; LIU, 2018, p. 3)

  The theories of Chomsky about language and mental grammar were the base for the approach on models NLP at that time. This model revolved around finite-state grammars, hand-crafted templates and ontologies. These approaches are highly interpretable and accurate, however, they are extremely limited to the domains they were modeled for, as are derived from expert knowledge and not from general data.

- **Empiricism:** this wave departed from the opposite view of language:

> "In contrast to rationalist approaches, empirical approaches assume that the human mind only begins with general operations for association, pattern recognition, and generalization. Rich sensory input is required to enable the mind to learn the detailed structure of natural language. (DENG; LIU, 2018, p. 4)

The developments therefore are "characterized by the exploitation of data corpora and of (shallow) machine learning, statistical or otherwise, to make use of such data" (DENG; LIU, 2018, p. 4) Algorithms such as Bayesian networks, support vector machines, decision trees, and, for neural networks, backpropagation algorithm were developed to extract information from large corpora, instead of relying on expert knowledge, which resulted that "the NLP systems, including speech recognition, language understanding, and machine translation, developed during the second wave performed a lot better and with higher robustness than those during the first wave" (DENG; LIU, 2018, p. 6).

- **Deep Learning:**

  The algorithms from the second wave were still far from human performance and the shallow ML models were not able to absorb large amounts of training data, which resulted in new models under the umbrella of deep learning. Deep learning models "exploit the powerful neural networks that contain multiple hidden layers to solve general machine learning tasks dispensing with feature engineering." (DENG; LIU, 2018, p. 7)

  Deep learning approaches are used successfully in several NLP tasks (as already mentioned in a previous section), and will be the focus on this work.

2.3.1 Levels of Linguistic information application

Machine learning models (being shallow or deep) makes use of varied information, linguistic or not. We will review now the types of linguistic information existent at each level. Since speech processing is not relevant for this paper and since we are dealing with textual data, we will skip this domain from now on.

2.3.1.1 Morphology

Textual data is continuous: a sequence of characters. However, language is symbolic, so that and language structures can be decomposed into tokens, which are mapped to words (which can have lexical or functional nature) and other structures, such as punctuation, spaces, etc.

Words in their turn can be further decomposed into meaningful units called morphemes (HASPELMATH; SIMS, 2013). For instance, suffixes, prefixes, stem, lemma, are all morphemes, meaning that they are constituents of words. In this domain, we classify words in classes, such as nouns, verbs, prepositions, conjunctions, articles, etc, based on the function in the sentence, and their semantic role. For instance, nouns usually represent entities, while verbs usually represent actions.

2.3.1.2 Syntax

Syntax goes a level up and studies how languages group words to form sentences. For instance, word order is a characteristic that varies intra and inter languages and may convey different meanings. Also, words have shared and lexical characteristics.

For instance, while at a word level, we have word classes such as prepositions, conjunctions, nouns, at a sentence level we have constituents such as subject, predicate, object, agreement. These are hierarchical information and the sentence level information is dependent on the word-level information. As an example, only words belonging to specific classes can occupy the role of subject, and their hierarchical order is fundamental.

As an example, given the following X-bar decomposition presented in Figure 1:

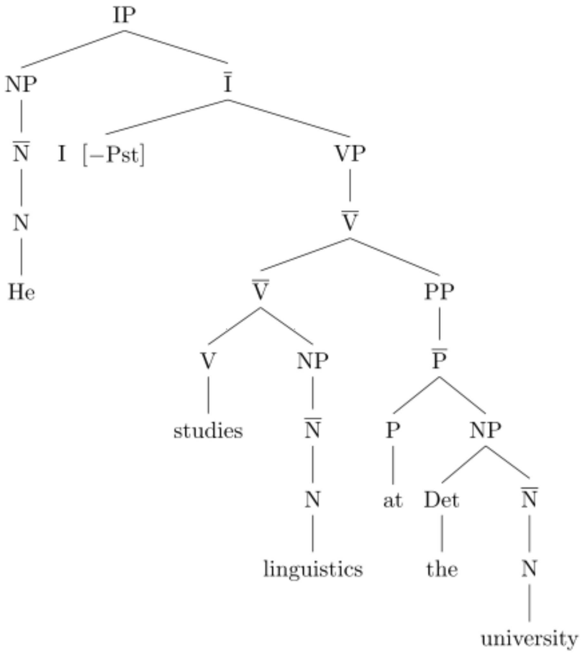


Figure 1 – Decomposition of constituents (X-bar notation)

The Noun Phrase *He* cannot be replaced by a verb or adverb. This is performed by the syntax module of the grammar. In the same way, the grammar knows the agreement rules of the language and conjugate the verb *studies* accordingly to the Noun Phrase (NP) to which it refers. These characteristics are frequently used in NLP and retrieved by using Parts-of-speech (POS) and dependency parsers.

2.3.1.3 Semantics

Morphemes form words, words form sentences, and in the end, meaning permeates and drives each level to accomplish the desired communication intent of the speaker. Semantics studies meaning (RIEMER, 2010) and the semantic knowledge that speakers have is structured in several characteristic and concepts.

As a matter of relevance for our methodology, we will cover some important concepts from semantics: sense, prototypes, hyponymy, synonymy, and frames.

Sense may be defined "as the general meaning or the concept underlying the word"(RIEMER, 2010, p. 17). However, some other factors may impact why a specific word is chosen or not for a particular communication instance. For example, consider situations where we want to be more informal or formal (registry) or situations where we want to be pejorative for effect. Speakers choose deliberately to say **cop** or **police officer** even though they point to the same concept. Likewise, there are situations where to say *kick the bucket* would be impolite, and the word *decease* or *pass away* would be preferred instead by the speaker.

This relates to the concept of synonymy, which is defined as a relation where "substitution of the definiens for the definiendum should be truth preserving in all contexts.", meaning that we can replace one word by its synonym without changing its meaning but eventually changing its registry, for instance. (RIEMER, 2010).

Another related concepts are hyponyms and hipernyms, which are relations of specificity versus general: "A standard identification procedure for hyponymy is based on the notion of class-inclusion: A is a hyponym of B if every A is necessarily a B, but not every B is necessarily an A." (RIEMER, 2010, p. 142). For example every car is a vehicle, but not all vehicles are cars (vehicles can be buses, trucks, etc). Therefore, car is a hyponym to vehicle and vehicle is a hypernym to car. Using WordNet, we are able to retrieve hyponyms and hypernyms for the words, which is incredibly useful for generalization, since it may improve the performance of neural network models. For instance, let us say we want to process a corpora of stolen reports:

1. Somebody stole my Fiesta in front my house yesterday.
2. I just took my Audi to the shop and then a guy pointed a gun at me and stole it.

If we query for *how many cars were stolen last year?*, the hyponymy relation may be able to generalize the different names of car to the concept. This is particularly important given that neural network models operate on dictionary base word vectors, so that each car name would be treated as a different word, preventing semantic generalization.

Prototype theory relates to the concept categorization. For instance, all vehicles usually have wheels and tires, resulting that "Categories (...) consist of entities with various shared attributes."(RIEMER, 2010, p. 230) Some entities may be more prototypical than others in the same category, for instance, a cat with no fur is less prototypical than a cat with fur for the [CAT] category, however, we can still recognize both entities as cats. Again, prototyping leads to generalization, however this type of information is much harder to incorporate into neural networks.

One of the ways is using Levin verb classes, which classifies verbs based on shared semantic characteristics, and it is available at VerbNet. (LEVIN, 1993). For instance, below we present one Levin class and the verbs it contains:

Verbs of Perception:

- see-30.1
- sight-30.2
- peer-30.3
- stimulus_subject-30.4

At last, frames or semantic frames relate to the theory of Frame Semantics proposed by (FILLMORE et al., 1976) within the framework of Cognitive Linguistics. This theory and its main concept, cognitive frame, can be summarized as follows:

> FRAME SEMANTICS is a research program in empirical semantics which emphasizes the continuities between language and experience, and provides a framework for presenting the results of that research. A FRAME is any system of concepts related in such a way that to understand any one concept it is necessary to understand the entire system; introducing any one concept results in all of them becoming available. In Frame Semantics, a word represents a category of experience; part of the research endeavor is the uncovering of reasons a speech community has for creating the category represented by the word and including that reason in the description of the meaning of the word. (PETRUCK, 1996, p.1)

As as example of a frame, extracted from the Framenet database, we have the frame Activity_finish, which contains the following **lexical units** (LU):

1. complete.v
2. completion.n
3. conclude.v
4. finish.v
5. graduate.v
6. tie up.v
7. wrap up.v

The lexical unit in this context is defined as: (..) "a pairing of a linguistic expression with a frame. Every lexical unit evokes a particular frame and can only be understood in relation to that frame." (IRMER, 2010, p. 190)

2.3.1.4 Pragmatics

Pragmatics is the study of the language as it is actually used in context. We will not include any pragmatic information in our analysis, therefore this topic will not be discussed.

2.3.1.5 Discourse:

Discourse studies related to a more abstract level of "being in the world, (...) integrate words, acts, values, beliefs, attitudes, and social identities as well as gestures, glances, body positions, and clothes" (GEE, 1989, p. 6) We will also not include any information at the discourse level at our analysis, so this topic will not be discussed.

2.3.2 Word Embeddings

This section will describe what word embeddings are and the most commonly used word embeddings in the NLP.

In NLP, especially when using in deep learning and neural network approaches, we need to represent (encode) text in a way that allows the text or words to be inputted to an neural network. Several techniques were proposed to accomplish this. More recently, the most common approach is "to represent words as dense vectors that are derived by various training methods inspired from neural-network language modeling (...) These representations, referred to as “neural embeddings” or “word embeddings”, have been shown to perform well across a variety of tasks (...) " (LEVY; GOLDBERG, 2014, p. 302).

Several techniques are used to generated word embeddings for pieces of text, such as: Bag of words (BoW), Term-Frequency - Inverse Document Frequency (TF-IDF), word2vec, GloVe, ELMo, and BERT. The first four approaches generate word vectors defined and context-free, meaning that the word representation is independent from the words in the context (words that come before or after the word to be encoded). They also generate static word embeddings, meaning that the same word in encoded in the same way regardless of its sentence.

ELMo and BERT, on the other hand, are contextual models, meaning that they generate a representation of each word that is based on the other words in the sentence, and also generate dynamic embeddings, meaning that that the word vector differs under the different sentences (WANG; CUI; ZHANG, 2019).

Contextual models can the further classified as unidirectional or shallowly bidirectional, and bidirectional or deeply bidirectional. Unidirectional means "that each word is only contextualized using the words to its left (or right)". ELMo is a shallowly bidirectional approach, which means that "ELMo uses the concatenation of independently trained left-to-right and right-to left LSTMs to generate features" (DEVLIN et al., 2018, p. 13).

BERT is defined as a deeply bidirectional model and generated dynamic embeddings, and has been used extensively in NLP tasks since its first publishing in (DEVLIN et al., 2018). Since BERT was not used by any research in the SemEval 2018-Task 7, and it is today one of the most prominent ways to generate word embeddings for semantic tasks, we will use this model to encode our corpus in one of the methods.

3 LITERATURE REVIEW

In this chapter we will review all related papers relevant for our research. From this analysis, we will be able to understand the models used for this relation extraction task and their performance.

3.1 Related Work

This research uses as corpora the SemEval 2018, Task 07 dataset (GÁBOR et al., 2018), and in this section we will analyze the related papers that used the same task and their approaches.

The dataset is separated into three subtasks: relation categorization in clean data (subtask 1.1), relation catgorization in noisy data (subtask 1.2), and relation extraction and categorization (subtask 2). Since the subtask 1.1 had the higher number of participants and published papers, we will focus on this subtask in this research.

To select the relevant papers published for this task, we have applied the following selection criteria:

1. We selected three important research repositories: IEEE Explore [1], ArXiv [2], and Google Scholar[3].

2. For each repository, we used the search query: "SemEval-2018" "Task 7" and filtered by year 2018 on.

3. Then, we manually went through all entries and read the abstract to determine whether the dataset used was the SemEval-2018, Task 7. Papers that used a different dataset were removed.

4. At last, duplicated entries, surveys, and papers written in languages other than English were removed. Some papers were duplicated because they described the same model and were published by the same authors, even though the papers themselves were written with some differences. We excluded them also.

After the selection outlined above, seventeen papers were read and analyzed. They will be summarized below, considering the following information: the features used, the methods (either deep learning or not), if they used external resources such as WordNet as features, and the F1 score that they accomplished in the task.

For the sake of brevity given the number of publications, the analysis of the features and methods will be focused on finding the similarities and gaps, therefore we may not include some specifics of each study.

[1] `https://ieeexplore.ieee.org/`

[2] `https://arxiv.org/`

[3] `http://scholar.google.com/`

In addition, the publications were numbered to aid the visualization in the tables below in the following way:

1. (LUAN; OSTENDORF; HAJISHIRZI, 2018)
2. (NOORALAHZADEH; ØVRELID; LØNNING, 2018)
3. (RENSLOW; NEUMANN, 2018)
4. (LENA et al., 2018)
5. (MACAVANEY et al., 2018)
6. (ROTSZTEJN; HOLLENSTEIN; ZHANG, 2018)
7. (DHYANI, 2018)
8. (GUO et al., 2019)
9. (JIN et al., 2018)
10. (PRATAP et al., 2018)
11. (KEIPER et al., 2018)
12. (YIN et al., 2018)
13. (MAHENDRAN; BRAHMANA; MCINNES, 2018)
14. (DRAGONI, 2018)
15. (BARIK; SIKDAR; GAMBÄCK, 2018)
16. (GLUHAK et al., 2018)
17. (SYSOEV; MAYOROV, 2018)

3.1.1 Features

The papers used a variety of linguistic features as input to the neural models or traditional classifiers. The Figure 2 displays the features and the number of papers that used them:

We can see that the number of different features used is high. The most commonly used are syntactic information in the form of part-of-speech tags. After comes pre-trained embeddings. In the majority of papers, the types of word embeddings used are word2vec and GloVe.

In addition, several papers used a combination of several features (including linguistic and non-linguistic information), although the preferred linguistic information used as feature carried syntactic information (POS), instead of semantic information (such as VerbNet classes).

3.1.2 Models

Now considering the learning algorithms used, the figure 3 summarizes them for each paper.

The most used models are Convolutional Neural Networks (CNN), followed by Long Short-Term Memory (LSTM) architectures, Recurrent Neural Networks (RNN), and Support Vector Machines (SVM), being the latter is the most non-deep learning method used.

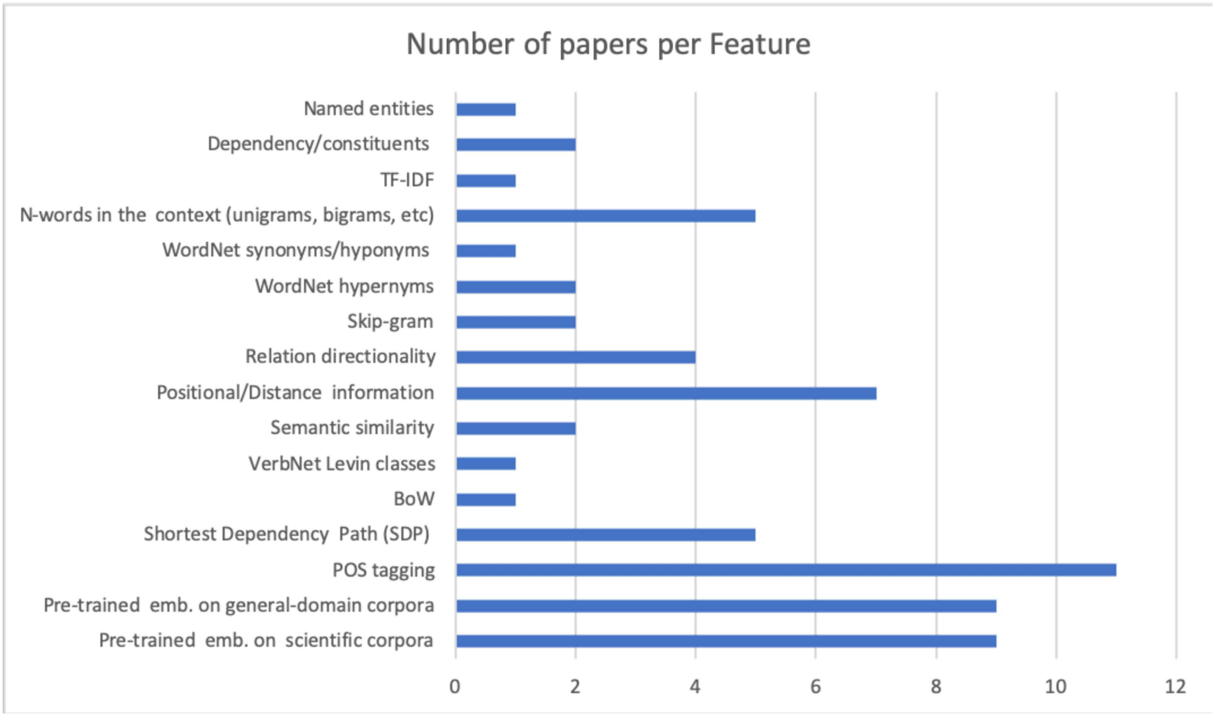


Figure 2 – Number of papers by type of feature

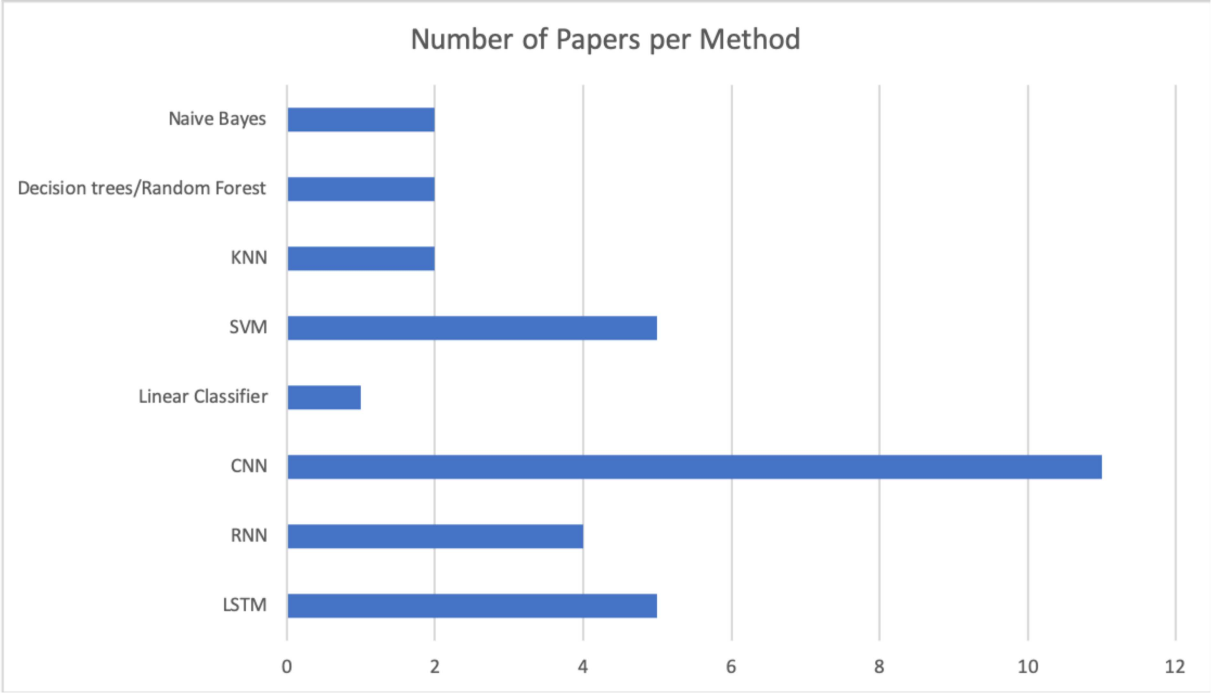


Figure 3 – Number of papers by type of method

3.1.3 External Resources

Regarding external resources, three papers used WordNet and one used VerbNet. From WordNet, the papers (JIN et al., 2018) and (PRATAP et al., 2018) used hypernyms, while (BARIK; SIKDAR; GAMBÄCK, 2018) used synonyms and hyponyms.

From VerbNet, the Levin classes were used as feature in (LENA et al., 2018).

3.1.4 Correlation of F1 and Features/Models

In this section, we will deep dive on the features and models used, especially analysing their performance given the F1 score of the algorithms.

In Table 1, the F1 scores and ranking of the methods are presented. Please note that the rankings defined here are defined between the papers reviewed, not the ranking in the SemEval 2018-Task 7 official submission, since we are considering only the models that had papers published.

Paper	F1 score	Ranking
6	81.7	1
1	78.9	2
2	76.7	3
4	74.9	4
10	74.2	5
9	72.7	6
16	69.7	7
17	64.9	8
8	64.59	9
5	60.9	10
12	49.1	11
7	48.1	12
15	47.4	13
11	44	14
3	39.9	15
13	20.3	16
14	18	17

Table 1 – Macro-F1 scores of the models from literature

The table 1 shows the breakdown of the F1 score for each method and their ranking. We can see that the performance of the models varied greatly. We will analyze if there is any correlation between the F1 scores and the models and features used.

In figure 4 presents the relation between the model used and the F1 score. We can see that there is a clear correlation in that deep learning models achieved better performance than non-deep learning models. This was taken into consideration when we chose the model for our experiments.

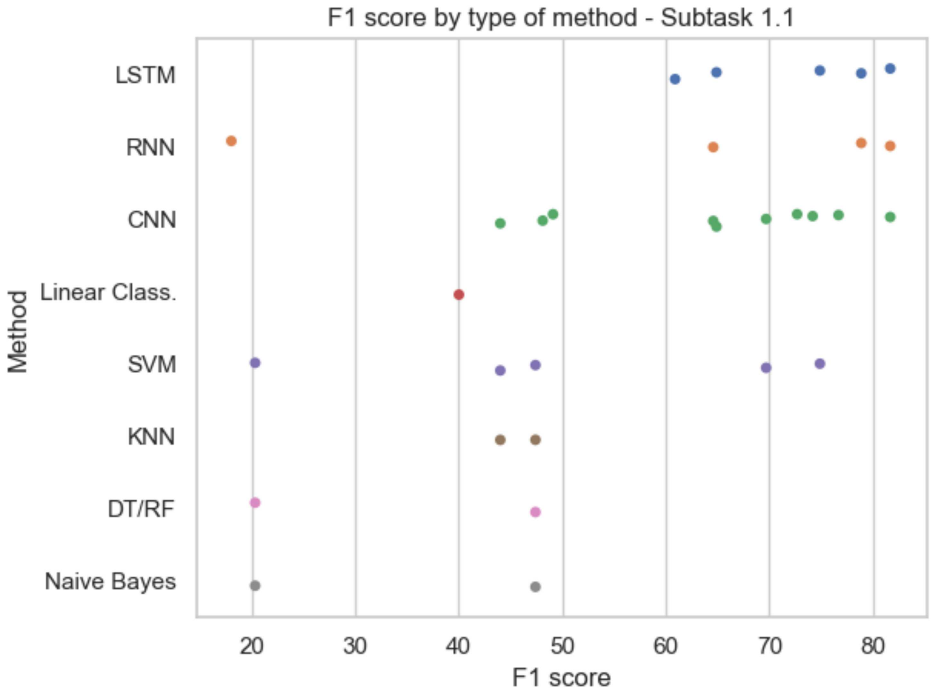


Figure 4 – F1 scores for each method

In figure 5 presents the relation between the features used and the F1 score. We can see that here, as opposed to the model, there is no clear relation between the features and the performance of the algorithm. Most of the features were used by algorithms which had quite different F1 scores, which bases the fact that it is not clear the impact that the features holds in the performance of the model, and from which we can understand that each feature is not capable of yielding consistently high scores when used independently.

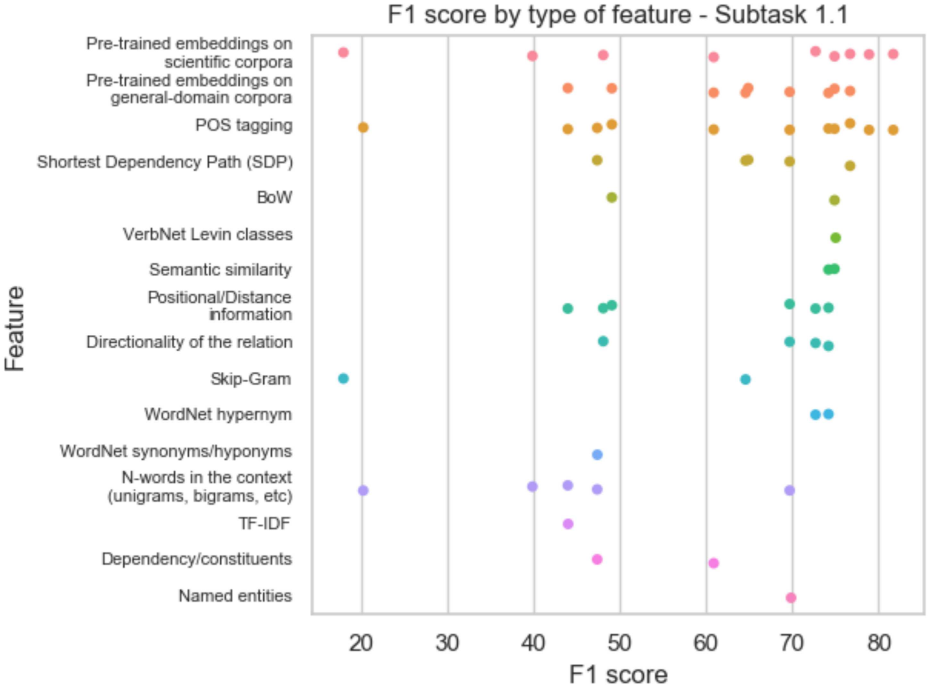


Figure 5 – F1 scores for each feature

Lastly, the summary of the word embeddings algorithms is presented in Table 2. Since the majority of the papers used word2vec or an equivalent technique, there is no enough data to analyze a possible relation between the F1 score and the type of word-embeddings.

Paper	Word embedding algorithm
1	word2vec
2	word2vec
3	word2vec
4	word2vec
5	fastText
6	word2vec
7	modified word2vec (dependency-based)
8	skip-gram word2vec
9	skip-gram word2vec
10	word2vec
11	GloVe
12	word2vec
13	Not specified
14	word2vec
15	Not specified
16	word2vec
17	fasttext

Table 2 – Word Embeddings Techniques for Each Model

3.2 Limitations from current state-of-the-art models

In this section, we will discuss the limitation derived from the literature review presented in a previous section.

As we could see from previous sections, the deep learning architectures achieved the highest scores in the SemEval 2018-Task 7 dataset. However, several distinct DL models had close F1 scores, even though they used different architectures and linguistic information as features. To further analyze the relation between the scores and the models and features used by the models, we will analyze the 6 top ranked methods and the features used. The Table 3 summarizes this information:

Paper	**6**	**1**	**2**	**4**	**10**	**9**
F1 score	81.7	78.9	76.7	74.9	74.2	72.7
Method	RNN/LSTM, CNN	RNN/LSTM	CNN	C-LSTM, SVM	CNN	CNN
Pre-trained embeddings on scientific corpora	X	X	X	X		X
Pre-trained embeddings on general-domain corpora			X	X	X	
POS tagging	X	X	X	X	X	
Shortest Dependency Path (SDP)			X			
VerbNet Levin classes				X		
Semantic similarity				X	X	
Positional information	X				X	X
Relation directionality					X	X
WordNet hypernyms					X	X

Table 3 – Summary of the Features for the Highest Scoring Models

The highest ranked architectures used few linguistic inputs and much more complex architectures. For instance, for the hightest ranked, the only purely semantic input was the pre-trained word embeddings in the task domain.

The other features carried only purely syntactic/distributional information (POS tags and distance given as integers from the entities). However, the other architectures had close F1 scores (less than 10 points), and they used very different linguistic data, including deep semantic information, such as hypernyms and Levin classes.

Given the above, it is not clear from the literature what exactly is the driver of the good or bad results in terms of relation extraction. With very distinct deep-learning architectures and linguistic information from distinct natures used as input, we can not conclude if the results have achieved a true generalization.

Another point of interest is that linguistic information from different dimensions (syntactic, semantic, distributional) were used, however their impact on the generalization of the model was not clear in the papers, especially considering that the benchmark of the task was the manual extraction and classification. Human language processing of written text comprises of information at various levels, e.g. morphology, syntax, semantics, pragmatics, lexicon, discourse, and very few of these levels were leveraged in the architectures, so the papers were not able to demonstrate the impact of each linguistic information in the performance of the architecture, or whether the difference between the neural network results compared to the human benchmark were due to a lack of linguistic information to reason with.

At last, the language representation models used by all the papers were word embeddings coded either with word2vec or GloVe, which are context-free models. However, recently, other types of representations such as ElMo (PETERS et al., 2018) and BERT (DEVLIN et al., 2018) have been made available. They are contextualized word representations and have been shown to outperform previous representations in a number of NLP tasks. These new models were released in 2018, and were not explored in the relation extraction task, given that the SemEval 2018 happened in the same year, which may prove to have potential when applied to this relation extraction task.

4 METHODS AND EXPERIMENTS

This section presents the dataset chosen for the relation extraction task and the proposed methodology.

We are using a benchmark dataset for relation extraction with annotated entities and semantic relations categorized into five categories, which were tailored specifically to be useful in extracting knowledge from scientific text.

After reviewing the literature, our research proposes to use a LSTM network and several different linguistic inputs, varying from many levels of linguistic abstraction (morphological, syntactic, and semantic). In addition, one of our main objectives is to understand exactly the most relevant linguistic information for a task such as relation extraction.

We intend to fill a gap in the literature by analyzing the impact that different types of linguistic information has on a neural network when applied to a semantic task.

4.1 Corpora

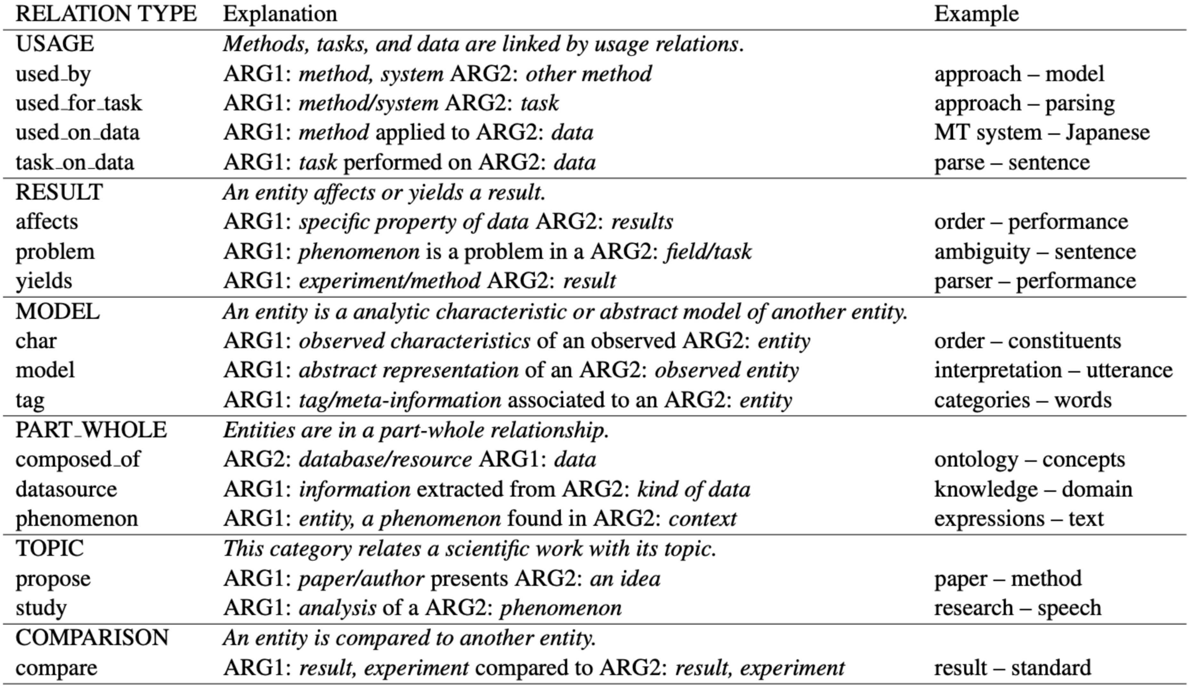
The dataset is the SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers (GÁBOR et al., 2018).

The task consists of identifying and extraction semantic relation between entities from abstract extracted from a scientific corpus. The training data contains 350 annotated complete abstracts for each subtask, with the corresponding relation instances and their relation categories. The test data consists of 150 complete abstracts. The research papers from where the abstracts were extracted are in the area of computational linguistics. As already cited in the literature review, since the subtask 1.1 had the higher number of participants and published papers, we will focus on this subtask in this research.

This subtask consists of categorizing each relation between two entities into 6 categories. The entities were manually annotated, as well as the relation instance and types. For this task, the information available is:

- **Training data:** manually annotated entities, manually annotated semantic instances and categories.
- **Test data:** manually annotated entities and manually annotated semantic instances.

The files were made available in XML files. The semantic relation topology is presented in Figure 6:



RELATION TYPE	Explanation	Example
USAGE	*Methods, tasks, and data are linked by usage relations.*	
used_by	ARG1: *method, system* ARG2: *other method*	approach – model
used_for_task	ARG1: *method/system* ARG2: *task*	approach – parsing
used_on_data	ARG1: *method* applied to ARG2: *data*	MT system – Japanese
task_on_data	ARG1: *task* performed on ARG2: *data*	parse – sentence
RESULT	*An entity affects or yields a result.*	
affects	ARG1: *specific property of data* ARG2: *results*	order – performance
problem	ARG1: *phenomenon* is a problem in a ARG2: *field/task*	ambiguity – sentence
yields	ARG1: *experiment/method* ARG2: *result*	parser – performance
MODEL	*An entity is a analytic characteristic or abstract model of another entity.*	
char	ARG1: *observed characteristics* of an observed ARG2: *entity*	order – constituents
model	ARG1: *abstract representation* of an ARG2: *observed entity*	interpretation – utterance
tag	ARG1: *tag/meta-information* associated to an ARG2: *entity*	categories – words
PART_WHOLE	*Entities are in a part-whole relationship.*	
composed_of	ARG2: *database/resource* ARG1: *data*	ontology – concepts
datasource	ARG1: *information* extracted from ARG2: *kind of data*	knowledge – domain
phenomenon	ARG1: *entity, a phenomenon* found in ARG2: *context*	expressions – text
TOPIC	*This category relates a scientific work with its topic.*	
propose	ARG1: *paper/author* presents ARG2: *an idea*	paper – method
study	ARG1: *analysis* of a ARG2: *phenomenon*	research – speech
COMPARISON	*An entity is compared to another entity.*	
compare	ARG1: *result, experiment* compared to ARG2: *result, experiment*	result – standard

Figure 6 – Semantic relation topology of the SemEval-2018-Task 7

Considering that we will be training a neural network with this dataset, it is important to note that the relation classes have data imbalance, as shown in the Table 4:

Relation	Training Frequency	Test Frequency
USAGE	483	175
TOPIC	18	3
MODEL-FEATURE	326	66
PART-WHOLE	234	70
COMPARE	95	21
RESULT	72	20
TOTAL	1228	355

Table 4 – Frequency of the relation classes

4.2 Data Processing

The corpus was pre-processed to normalize the text. The following pre-processing steps were performed:

- Context delimitation: for each entity pair with a relation we have delimited the sentence with the first entity, the second entity, and all words that occur between the two entities, what we will call context. Since the text corpus is composed of abstracts of varying lengths and words closer from each other are presupposed to impose a stronger constraint on their meaning (BROSSEAU-VILLENEUVE; NIE; KANDO, 2010), and considering we need to generate fixed length word vectors, we have chosen to delimit the sentences in this way to generate the word embeddings and train our models.

- Tokenization: separate the text into tokens that correspond to words;
- Lemmatization: replace the words with their lemmatized form, which is the dictionary form. For instance, verbs are mapped to their infinitive form, plural nouns are mapped to their singular form, etc.
- Punctuation removal.

To perform the tokenization and lemmatization, we used the freely available library spacy[1] in Python.

4.3 Model

In this section we will present the model used for the experiments. We will detail the deep learning architecture, the hyperparametrization, the feature engineering, and at last the experimental setups.

4.3.1 Deep Learning Architecture

Considering the literature review and the analysis derived from it, we propose to use a deep learning method that presented the best results: a Recurrent Neural Network (RNN) with an LSTM architecture. This is driven by the facts that firstly, this architecture yielded the best results in the SemEval-2018-Task up to the moment this paper is being written (both first and second best results used this model), and secondly, RNN/LSTM models have been used with successful results in many NLP tasks, together with RNN (YIN et al., 2017).

Convolutional neural networks (CNN) have also demonstrated great results, as seen in the literature review, and the best results used a combination of RNN and CNN. However, the main objective of this research is to use a simpler deep learning architecture and explore further the linguistic information given to the model, which is hypothesized to compensate for a less complex neural model, than proposing a new deep learning model.

The neural networks were implemented using the Keras Sequential model when inputting only one layer, or the Functional model when concatenating layers. For each experimental setup, the sizes of the layers change accordingly with the dimensionality of features.

The basic architecture contains one or multiple layers as input, one Bidirectional layer using LSTM with dropout rate equal to 20%, and a Dense layer with activation softmax with size 6, the number of possible relations.

[1] `https://spacy.io/`

4.3.1.1 Hyperparametrization and implementation

To implement the neural networks, we have used the library Keras built on top of TensorFlow in a Python implementation.

These values are:

- Batch size: 128
- Dropout rate: 0.2
- Validation split: 0.2
- Optimizer: adam
- Loss measure: categorical_crossentropy

In order to prevent *overfitting*, we have used *Keras EarlyStopping* as a callback with epochs set to 100, and monitor set to val_loss. With this, the neural network will stop the learning when the validation_loss stopped improving on the dataset.

These parameter values were all based on the standard values used in the literature of deep learning.

4.3.2 The features

The main goal of this paper is to explore and validate the impact of different linguistic information in the model. The models reviewed earlier showed that many different dimensions and linguistic data were inputted in the most successful architectures in the relation extraction task, however the impact of each one was not further explored.

Considering this, we propose to explore the following linguistic information as features of our RNN model:

- **Morphological/Lexical/Distributional**
  - *N-grams:* the distributional aspects of the language are very important. The N-grams allows the contextualization of the words in what is called collocations (in Linguistics) or N-gram models (in Computational Linguistics). The underlying assumption is that words are not independent from one another, and there are sequences of words (n-grams or word-coocurrences) in the languages that occur with greater frequency than other sequences. (EVERT, 2005)
  - *Empty Words*: words belonging to closed-ended classes that usually do not carry much semantic information. However, they may be useful to identify relations such as PART-WHOLE, which may have the form "<entity> of <entity>", as in the excerpt "database of words".

- **Syntactic**
  - *Part-of-speech (POS) tagging:* refers to identifying the morpho-syntactic class of each token in the sentence (example, noun, verb, adjective).
  - *Dependency tagging:* refers to identifying the syntactic role of each token in the sentence (subject, predicate, modifier).
- **Semantic**
  - *Synonyms, Hypernyms, Hyponyms:* parts of speech can be divided into two high-level categories: closed class and open class, where the four open classes are: nouns, verbs, adjectives, and adverbs. (MARTIN; JURAFSKY, 2009) Open class words carry more semantic meaning than closed class words (which play a more functional role), and they have semantic relations with words from the same class, such as hypernym, hyponym, and synonym.

    Lexical Synonymy refers to the relation between two words having the same sense (identity meaning), or in a broader sense "two expressions have the same contextual effect is what justifies labelling the substituted words as synonyms in that context.". (RIEMER, 2010, p. 151)

    Hyponymy refers to the hierarchical relation between in which "A standard identification procedure for hyponymy is based on the notion of class-inclusion: A is a hyponym of B if every A is necessarily a B, but not every B is necessarily an A" (RIEMER, 2010, p.142)

    Hypernymy refers to the same hierarchical relation as hyponymy, but denotes the opposite semantic term. While hyponym is the lower term of the hierarchy (more specific), hypernym (or hyperonym) is a hyponymic hierarchy (RIEMER, 2010). For instance: musical instrument and stringed instrument are both hyperonyms of violin, while violin is a hyponym of musical instrument and stringed instrument.

    A well-known database of hyponyms, synonyms, and hypernyms is available: WordNet. We propose to use WordNet to explore the impact of word-based semantics in the relation extraction task. This resource was used in the papers from the literature review, however, it is not clear from the papers whether using this resource actually improved the relation extraction and classification. (MILLER, 1995)
  - *Contextualized Word Embeddings*: as discussed in the literature review, two new contextual models of language representations were published since 2018: BERT (DEVLIN et al., 2018) and ELMo (PETERS et al., 2018). They differ from the traditional word models as they are context-sensitive embeddings, while BoW, word2vec, and GloVe are context-free representations. (ALSENTZER et al., 2019) BERT in specific "has, in general, been found to be superior to ELMo and far superior to non-contextual embeddings on a variety of tasks (...) " (ALSENTZER et al., 2019,

p. 2) and has not been used in any of the models from the literature applied to the SemEval-2018-Task 7. Considering this, we propose to use this representation in the relation extraction task, replacing the traditional non-contextual embeddings (word2vec, Glove, etc). Also, we propose to use this representation in the pre-trained corpora to be used by the model. Whether pre-trained models perform better with general or domain-specific embeddings in the relation extraction task was not answered by the literature review, therefore we intend to investigate this further.

- *FrameNet:* is a database of semantic frames for English (BAKER; FILLMORE; LOWE, 1998), and it is based on the Frame Semantics theory. According to (JOHNSON; FILLMORE, 2000, p. 56):

> Frame semantics characterizes the semantic and syntactic properties of predicating words by relating them to semantic frames. These are schematic representations of situations involving various participants, props, and other conceptual roles, each of which is a frame element (FE).

- *Levin classes:* is a system of verb classification proposed by (LEVIN, 1993) that groups verbs by shared semantic characteristics. VerbNet[2] is a public database that allows us to query verb classes using lemmas.

4.4 Word Embeddings

For the experiments, we will use four types of word embeddings:

1. **Tokenizer:** from Keras[3] library, it creates vectors with the integer representation of the words. This is a fast and simple way to create vectors using static dictionaries trained in the data. This technique was chosen because it is the simplest way of creating vectors from words and sentences. In addition, the Tokenizer is trained only in the dataset itself, no pre-trained embeddings are used.

2. **doc2vec**: implementation of word2vec for documents. We are using the library gensim to train and generate the sentence embeddings. This technique was chosen because it was the most used approach in the reference literature and it is a common approach used to generate word embeddings. In addition, as with the Tokenizer, no pre-trained embeddings are used.

3. **GloVe**[4]: generates static word vectors using pre-trained word vectors. We are using spacy

[2] https://verbs.colorado.edu/verbnet/

[3] https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer

[4] https://nlp.stanford.edu/projects/glove/

to generate the GloVe vectors based on the model en_core_web_lg [5], which is a "English multi-task CNN trained on OntoNotes, with GloVe vectors trained on Common Crawl."

4. **BERT**:[6] generates contextual and dynamic word embeddings using a pre-trained model. To facilite the generation of the word embeddings, we are using the implementation by bert-as-a-service[7].

   We are using the pre-trained model *BERT-Base, Uncased*, which has 12-layers and 768-hidden states per layer. BERT accepts 1 or 2 sentences, and in order to get the encodings for the sentences, each sentence or sentence pair needs to be masked with CLS (begin of sentence) and SEP (between sentences and end of sentences). For example:

   'The cat is sick'

   The input for BERT is: [CLS] 'The' 'cat' 'is' 'sick' [SEP]

   Additionally, we have configured the service in the following way (all other values are default):

   (a) **max_seq_len**: 40. Maximum size of our sentence in tokens.

   (b) **pooling_strategy**: REDUCE_MEAN (default). This will generate a vector of 768 positions for a sentence.

   (c) **mask_cls_sep**: True. Masks with zeros the CLS and SEP tokens.

4.5 Linguistic Information

For each type of word embedding, we will train neural networks with only the word embeddings, and also with the word embeddings together with different types of linguistic information:

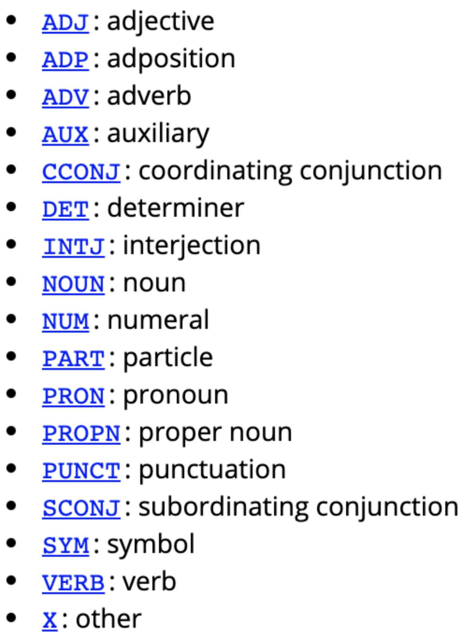
1. **Part-of-Speech tags and Dependency Tags**: we used the spacy library to generate tokens and tag the corpus with part-of-speech (POS) and dependency (DEP) tags.

   The POS tags represent the morphological class which token belongs to, for instance, verb, noun, pronoun, etc. The tags follow the Universal Dependencies v2 POS tag set, presented in Figure 7:

[5] https://spacy.io/models/en#en_core_web_lg

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

[7] https://github.com/hanxiao/bert-as-service



- ADJ: adjective
- ADP: adposition
- ADV: adverb
- AUX: auxiliary
- CCONJ: coordinating conjunction
- DET: determiner
- INTJ: interjection
- NOUN: noun
- NUM: numeral
- PART: particle
- PRON: pronoun
- PROPN: proper noun
- PUNCT: punctuation
- SCONJ: subordinating conjunction
- SYM: symbol
- VERB: verb
- X: other

Figure 7 – Part-of-speech tag set

The dependency tags annotate the syntactic relation between the tokens in the sentence, such as nominal subject, object, determiner, etc. The tags follow the ClearNLP Dependency Labels tag set.[8]

2. **Hypernyms**: The hypernyms were obtained for each token using the WordNet database. For each tokken, we used the lemma form of the token to query WordNet to retrieve all synsets, using the POS tag as additional parameter. We picked the first synset and queried the hypernyms. From the list of hypernyms, we have picked the first of the list and replaced the original token of the text with the hypernym. If no synset or hypernym was available in WordNet, we kept the original token.

3. **Semantic Frames**: To retrieve the semantic frames, we have used FrameNet database, using the interface available in the Natural Language Toolkit (NLTK).[9] This processing was performed for each lemma and the first frame found was used, replacing the token with the framename. If no frame was found, the original token was kept in the text.

4. **Verb classes**: To retrieve the verb classes, we have used the VerbNet database, using the interface available in the NLTK. For each verb in the sentence, the verb classes were retrieved and the token was replaced by the first class. Other non-verb tokens were kept as originally, as well as verb with no classes in the database.

[8] https://github.com/clir/clearnlp-guideline

[9] http://www.nltk.org/howto/framenet.html

4.6 Experiments and evaluations

Considering the features described in the previous sections, we have combined each different type of word embedding, as shown in Table 5:

Linguistic Feature	Word Embeddings			
Only embeddings	Tokenizer	word2vec	GloVe	BERT
POS + DEP	Tokenizer	word2vec	GloVe	BERT
Hypernyms	Tokenizer	word2vec	GloVe	BERT
Frames	Tokenizer	word2vec	GloVe	BERT
Verb classes	Tokenizer	word2vec	GloVe	BERT

Table 5 – Experimental setups

The models were evaluated using the following metrics: accuracy, macro F1 score, and weighted F1 score. Since the dataset had a high class imbalance, the weighted F1 score is more precise and will be used to compare the models in the analysis. To compare the models with the baseline models from the literature, we will use the macro F1 score, which was the metric used by the SemEval 2018-Task 7 scorer.

We have executed each setup three times and compared their F1 scores. Since the F1 scores did not vary significantly between runs, we have picked for each setup the higher score of the three executions.

5 RESULTS

In this section we will present and analyze the results of the experiments outlined in the previous sections.

5.1 Scores

For each of the proposed experiments, the accuracy, macro F1 score and weighted F1 score are presented in Table 6:

Word embeddings	Linguistic information	Accuracy	Macro F1 score	Weighted F1 score
Tokenizer	Only embeddings	0.63	0.47	0.63
	POS + DEP	0.44	0.17	0.38
	Hypernyms	0.65	0.50	0.65
	Frames	0.61	0.41	0.60
	Verb classes	0.62	0.46	0.62
word2vec	Only embeddings	0.65	0.43	0.64
	POS + DEP	0.59	0.40	0.59
	Hypernyms	0.60	0.38	0.59
	Frames	0.57	0.29	0.52
	Verb classes	0.65	0.40	0.62
GloVe	Only embeddings	0.65	0.45	0.64
	POS + DEP	0.63	0.45	0.63
	Hypernyms	0.63	0.42	0.62
	Frames	0.55	0.31	0.53
	Verb classes	0.64	0.44	0.63
BERT	Only embeddings	0.63	0.45	0.62
	POS + DEP	0.61	0.46	0.61
	Hypernyms	0.61	0.40	0.60
	Frames	0.54	0.38	0.53
	Verb classes	0.59	0.43	0.59

Table 6 – Experiment results

To analyze these results, we will divide the analysis into the following subsections: impact of different word embeddings, impact of syntactic information, impact of semantic information.

5.2 Impact of Word Embeddings

Considering the weighted F1 score, there is no significant difference between all the four types of word embeddings, as wen can see from the plot below:

The word2vec and GloVe techniques had a slightest higher accuracy than BERT and the tokenizer, however the difference is not significant. To understand that, we will now analyze the differences between them regarding the classification of each relation by the different types of word embeddings.

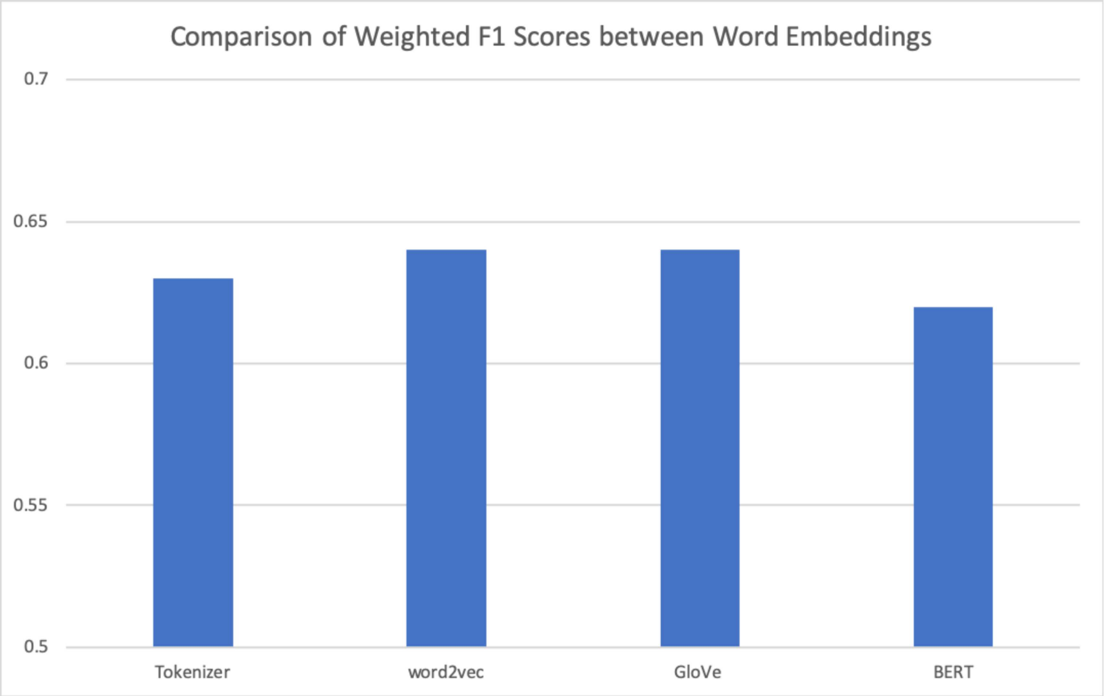


Figure 8 – Comparison of F1 score for the types of embeddings

In figure 8, we can see that there it no significant difference in performance for the different types of embeddings.

We can also see this when looking at the confusion matrixes for all the word embeddings. The confusion matrixes are presented in Figures 9, 10, 11, and 12.

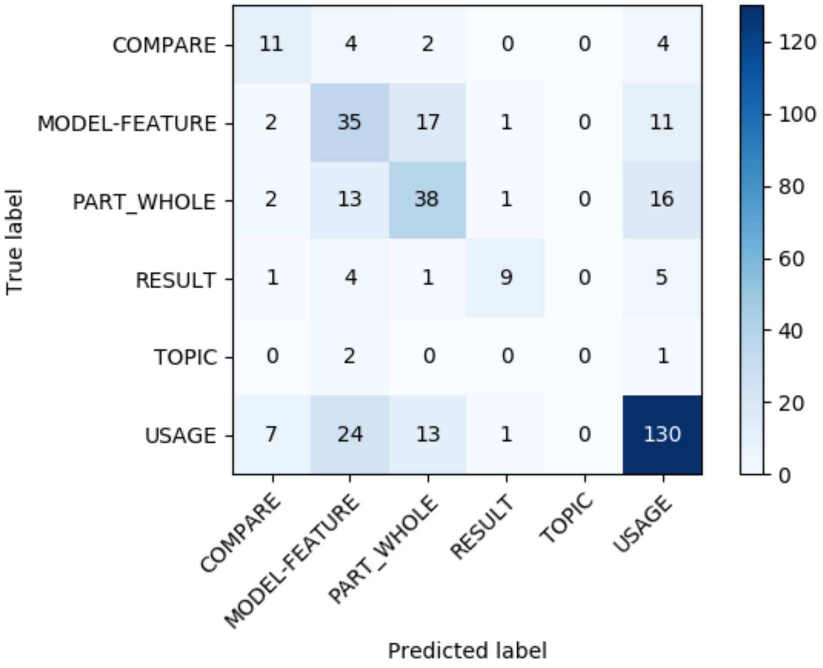


Figure 9 – Confusion Matrix for classifier using Tokenizer word embeddings

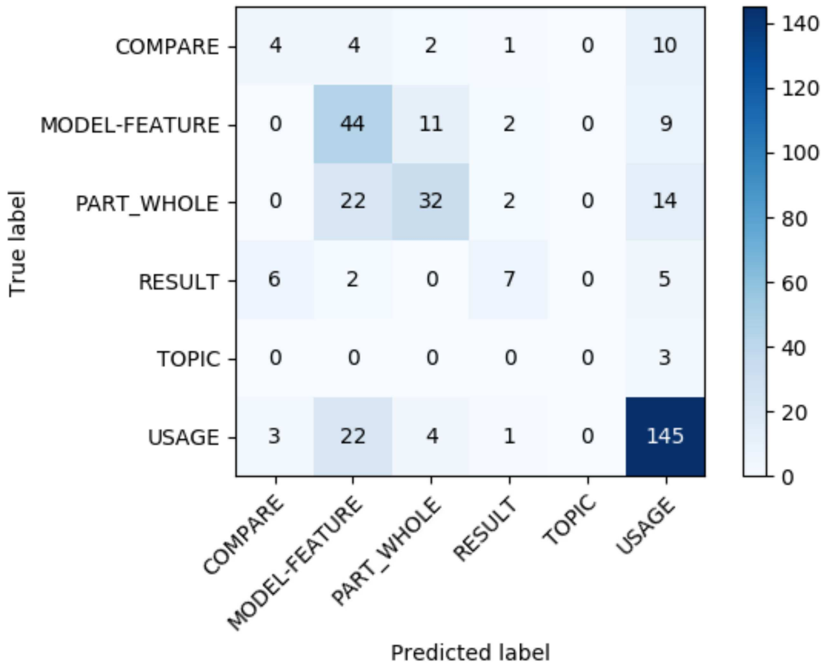


Figure 10 – Confusion Matrix for classifier using word2vec word embeddings

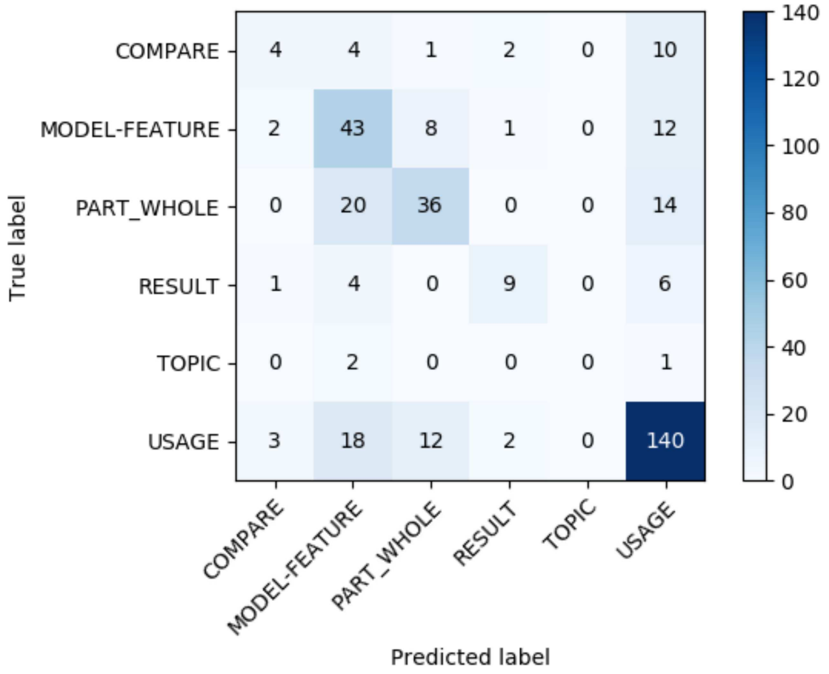
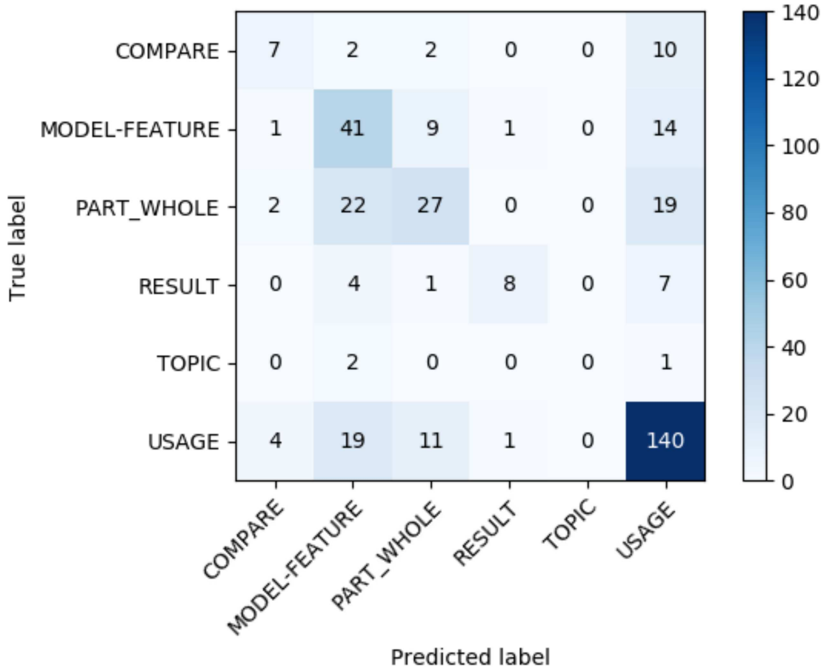


Figure 11 – Confusion Matrix for classifier using GloVe word embeddings



	COMPARE	MODEL-FEATURE	PART_WHOLE	RESULT	TOPIC	USAGE
COMPARE	7	2	2	0	0	10
MODEL-FEATURE	1	41	9	1	0	14
PART_WHOLE	2	22	27	0	0	19
RESULT	0	4	1	8	0	7
TOPIC	0	2	0	0	0	1
USAGE	4	19	11	1	0	140

Figure 12 – Confusion Matrix for classifier using BERT word embeddings

Even though the F1 scores are quite similar between the different word embeddings, the word embeddings type differed slightly in the number of correctly classified samples, but overall, the accuracy of classification of the USAGE relation was the best. Also, for all experiments, several instances of USAGE were incorrectly classified as PART_WHOLE or MODEL-FEATURE, and vice-versa, so there seems to be a overlap between these three classes.

To visualize the word embeddings, we have used the t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization technique proposed by MAATEN; HINTON, 2008. We use this technique to accomplish the dimensionality reduction of the feature, to be able to plot them in a two-dimensional graph. The figure 13 shows the visualization for all types of embeddings:

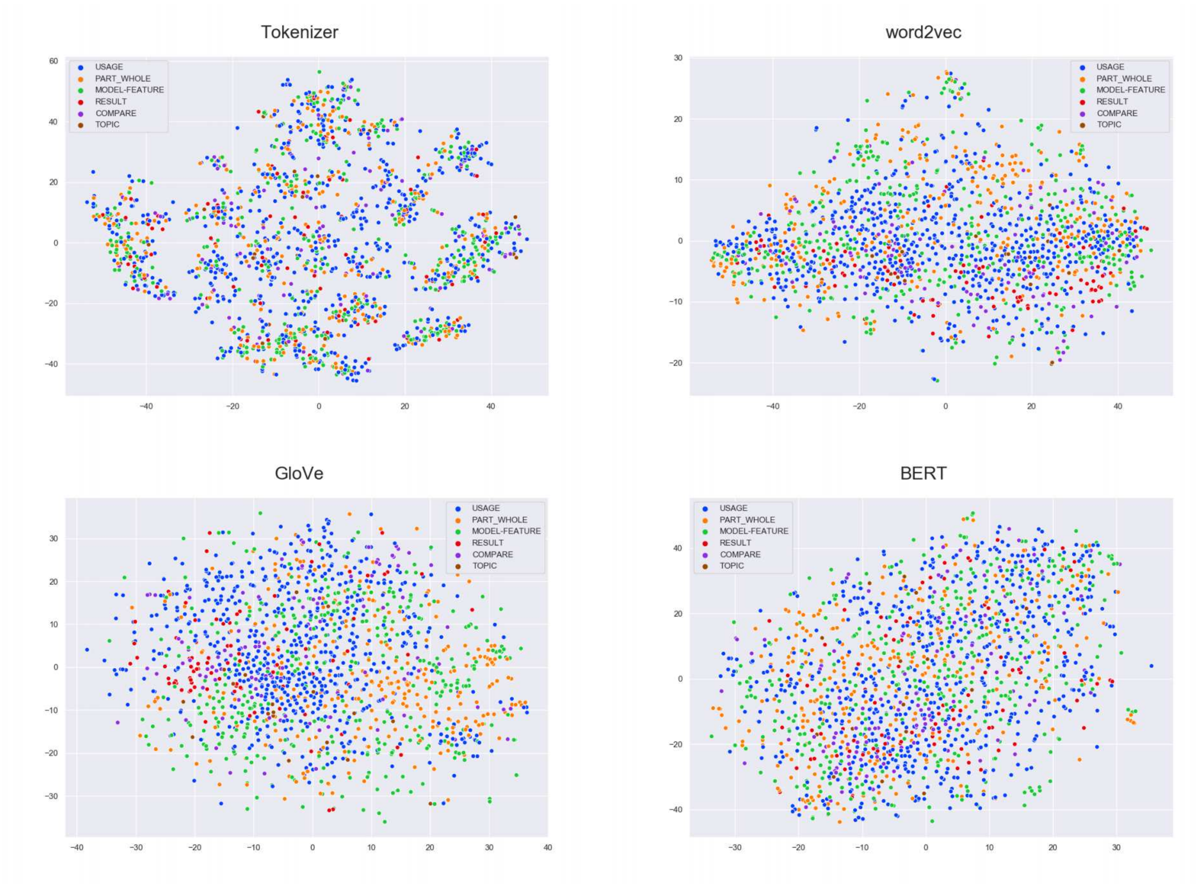


Figure 13 – t-SNE visualization for the word embeddings

As we can see from the Figure 13, the classes have a high degree of overlap, independently of the type of word embeddings, which explains why the F1 scores and accuracy did not vary between word embeddings. Also, explains why the the classes USAGE, MODEL-FEATURE and PART_WHOLE are misclassified, since these three classes have a higher degree of dispersion, and a high degree of overlap between them.

Apart from that, the models still had a good accuracy considering that if all the instances were classified with the relation with the highest number of samples (in this case, USAGE) which would be a model with no generalization, the macro F1 score would be 0.11 and the weighted F1 score would be 0.35. For our models, the F1 scores (both the macro and the weighted) were higher than that, which shows that the models were able to learn and classify correctly several relations and achieve a degree of generalization from the data. However, since the corpus itself has a high degree of overlap for the relation for all of the word embeddings that we used, the models were not able to classify correctly all instances. It is worth noting that the word embeddings techniques used range from the simplest (Tokenizer) to the most commonly used (word2vec and GloVe) up to the state-of-the-art word embeddings technique (BERT), which indicates that to achieve a very good performance for this relation extraction it is required more information that only the word embeddings from the text may provide.

5.3 Impact of Syntactic Information

In all experiments, the part-of-speech and dependency tags for the corpus were inputted to the neural networks together with the word embeddings. However, in all the experimental setups, this information did not improve the accuracy of the models, and for the model that uses a Tokenizer as word embedding technique, feeding the syntactic information to the model actually decreased significantly its accuracy.

The negative impact of the tags in the Tokenizer can be explained considering that the Tokenizer generated highly sparse and low-dimensional vectors (vector with size 100), which made it less robust to noise in the input. For the other models, the word embeddings are dense vectors with a higher dimensionality (vectors with size 300 for GloVe and word2vec, and size 768 for BERT).

To analyze why the syntactic information was not helpful for the model, we can use the t-SNE visualizations presented in Figure 14 and 15.

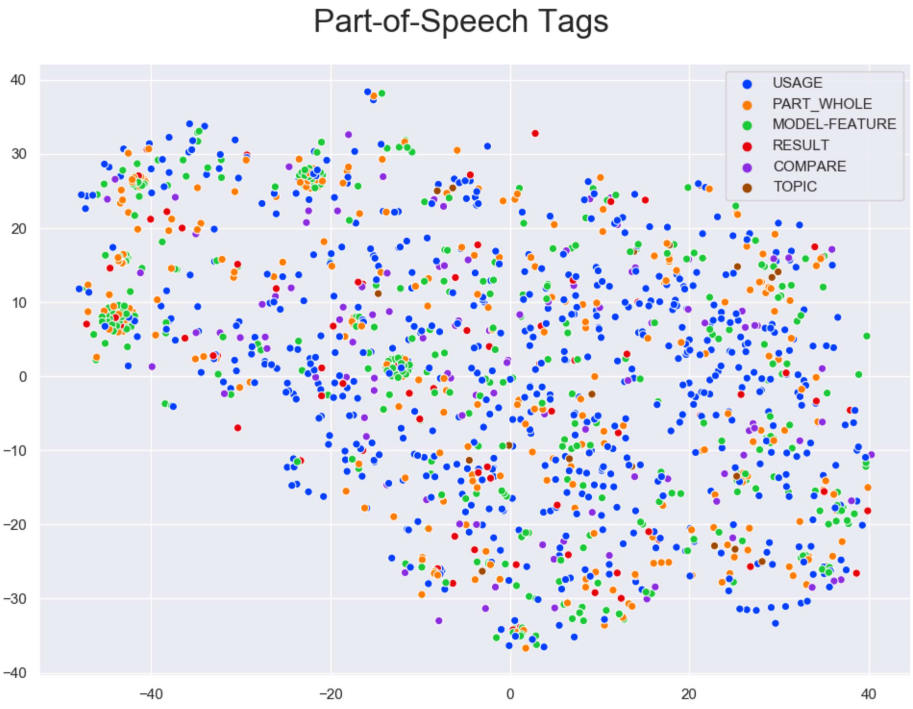


Figure 14 – t-SNE visualization for Part-of-Speech Tags

As we can see from the plots in Figures 14 and 15, there is no clear correlation between the relation to be classified and either the part-of-speech or dependency information. This is interesting considering that the model with highest score from the SemEval 2018-Task 7 used only word2vec and part-of-speech tagging as features, which do not seem to have a correlation to the relation classes, so these two types of information do not seem to be enough for a model to classify the relations. This also explain why adding the syntactic information to the our models

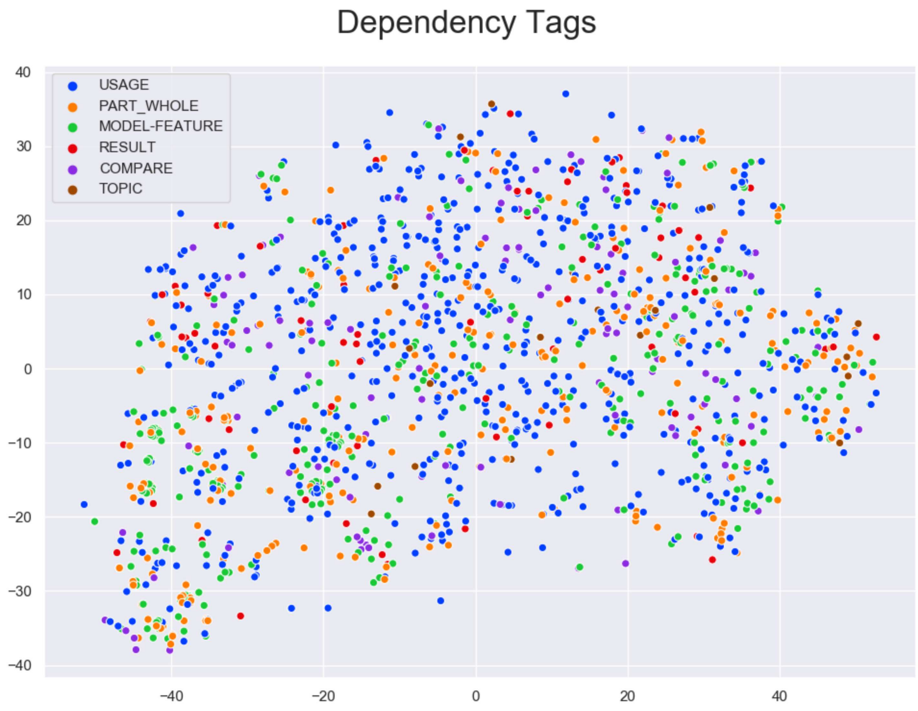


Figure 15 – t-SNE visualization for Dependency Tags

did not improve accuracy on this dataset.

5.4 Impact of Semantic Information

For all our models, the variance in F1 score for the different types of semantic information is presented in the plot in Figure 16.

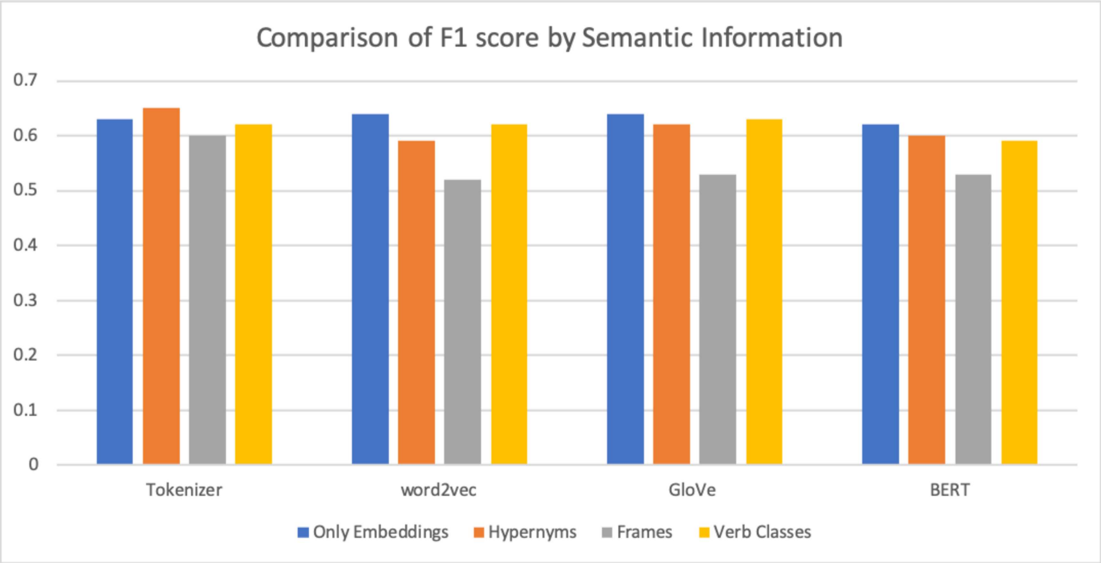


Figure 16 – F1 score comparison by Semantic Information

Overall, the semantic information did not contribute to improve the performance of the model. For most of the types of embeddings, the use of semantic information as features did not make any significant improvement. The F1 score using only word-embeddings was higher that using word-embeddings with enriched semantic data. For the model that uses the Tokenizer, the hypernyms contributed marginally to the improvement of the model performance, given that it may increase the genealization of the data, however in this dataset, this did not seem to have much effect, probably due to the size of the corpora.

The higher impact in the Tokenizer again may be due to the fact that this word embedding technique considers only the specific dataset, while the other word embeddings use pre-trained models to generate the embeddings for the corpus dataset. Because of this, using hypernyms in the Tokenizer models affects directly the word embeddings, while for the others, the embeddings depend more on the pre-trained corpus. Since the pre-trained embeddings are trained with general corpora (Wikipedia, etc), the hypernyms mapped for this dataset may not be present in the pre-trained model.

To further analyze the performance using different semantic informations, let us compare the same sentence with different semantic enrichments:

- **Original sentence:** stem model be base on statistical machine translation
- **Hypernyms:** originate_in hypothesis metallic_element base on statistical device written_record
- **Frames:** gizmo exemplar abounding_with building_subparts abandonment capability gizmo translating
- **Verb classes:** appear model be base on statistical machine translation

It is very clear that when using verb classes, the majority of the tokens remained the same as in the original sentence. For the hypernyms and semantic frames, there is a higher degree of substitution, since they apply to several morphological categories, not only to verbs.

We can see this clearer when comparing the number of unique tokens found for each type of semantic enrichment:

- **Original text:** 2116 unique tokens
- **Hypernyms:** 1742 unique tokens
- **Frames:** 1439 unique tokens
- **Verb classes:** 2030 unique tokens

We can see that using hypernyms or frames reduced the number of tokens, which could improve the generalization of the texts. However, only for the hypernyms this generalization had a slightly impact in the model performance. We have expected that enhancing the text with semantic information would make the different relations more distinguishable, providing the model with a more accurate information on how to predict the relations. However, we can again compare the visualization of the word embeddings for each type of semantic enrichment, as presented in Figures 17, 18, 19, and 20.

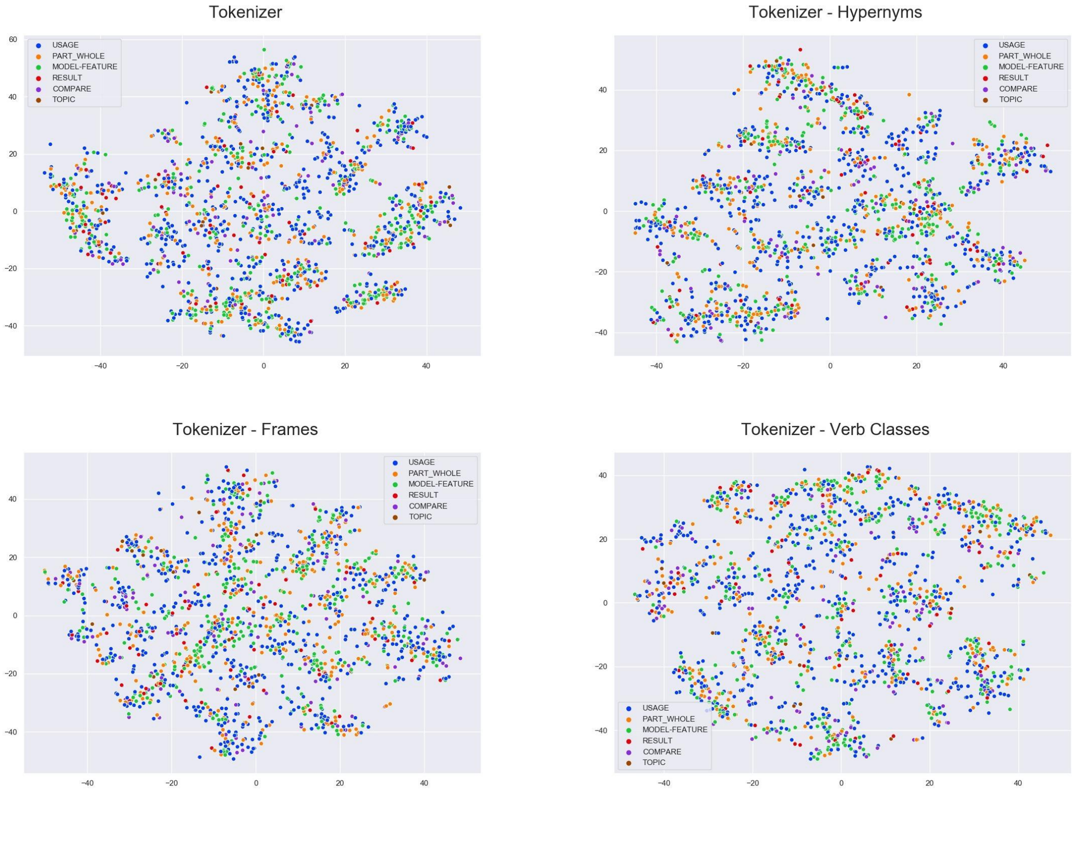


Figure 17 – Comparison of different semantic enrichments for Tokenizer embeddings

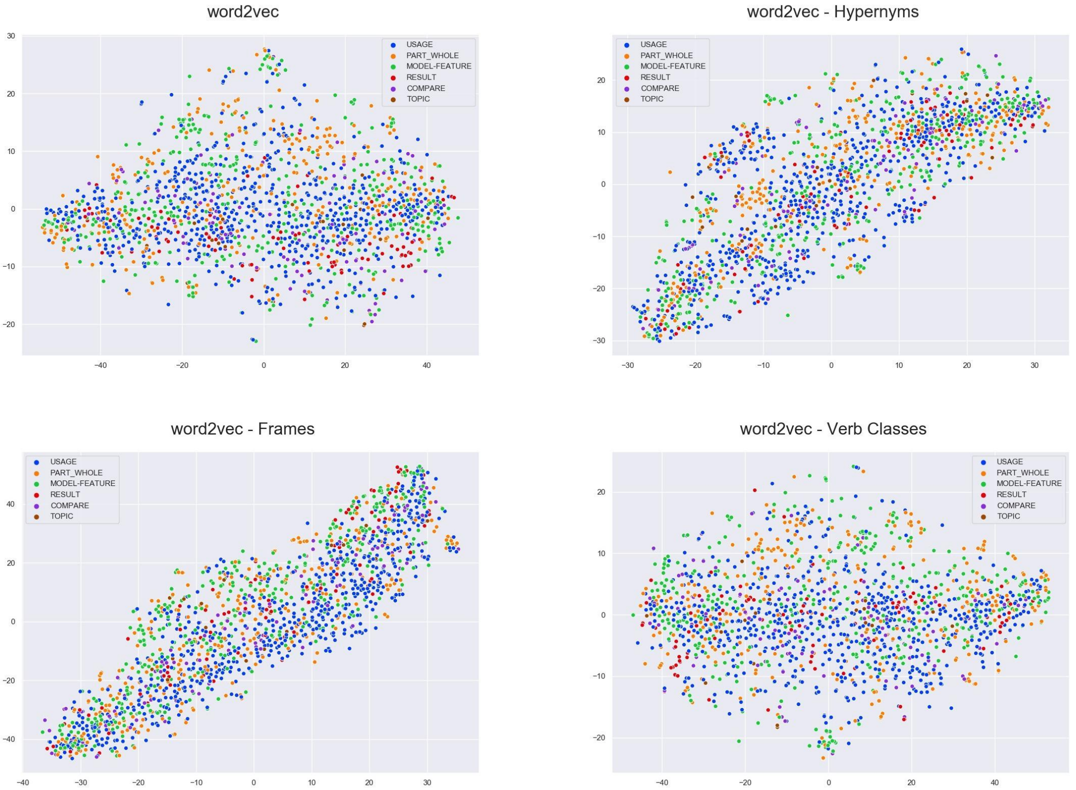


Figure 18 – Comparison of different semantic enrichments for Word2Vec embeddings

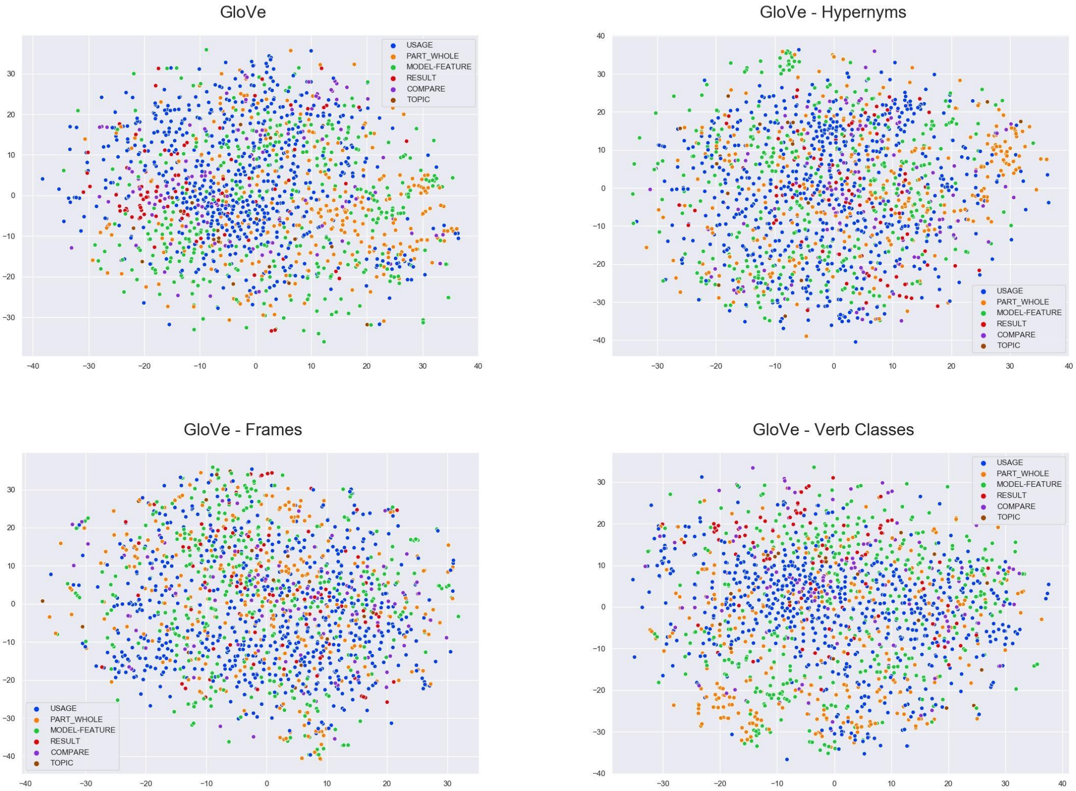


Figure 19 – Comparison of different semantic enrichments for GloVe embeddings

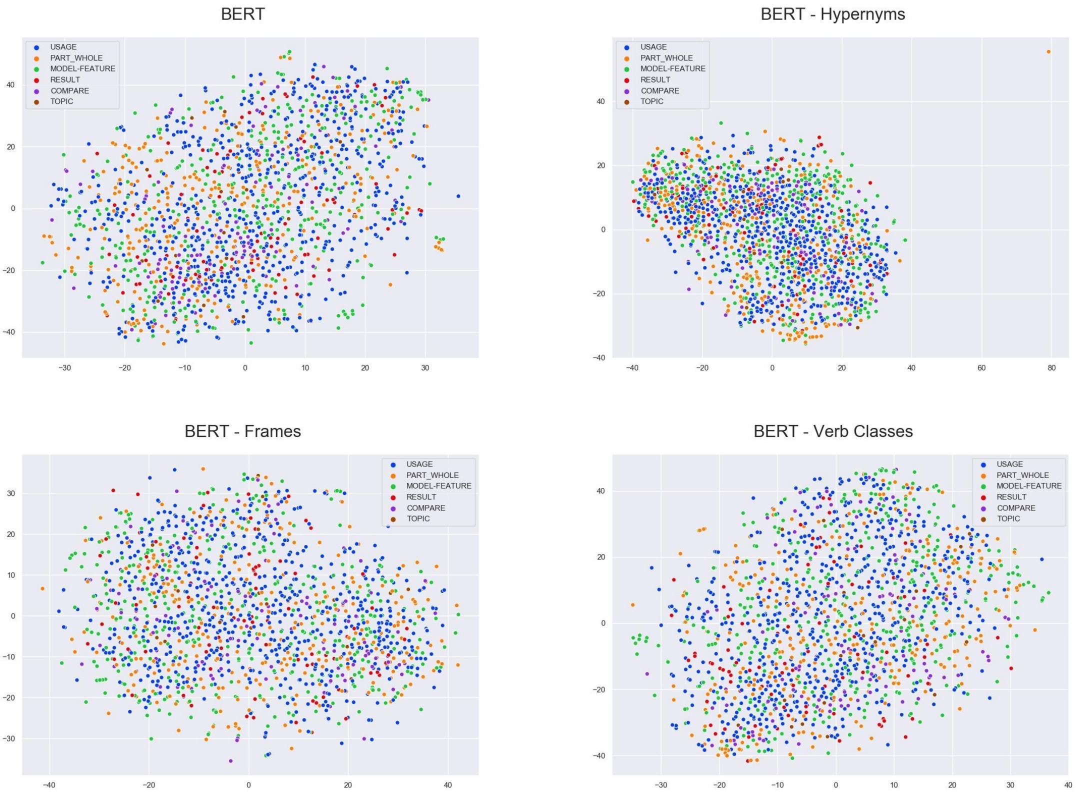


Figure 20 – Comparison of different semantic enrichments for BERT embeddings

We can see from the Figures 17, 18, 19, and 20 that there is not much improvement when using different types of semantic information, since the classes are still not very distinguishable from one another. For the word2vec embeddings, we can see a lower dispersion when the embeddings were generated with hypernyms and semantic frames, but the relations still remained overlapped, therefore the model was unable to achieve an improvement in the classification, given that there is no clear separation of the classes in the data.

From these experiments, we can conclude that for this specific task, using semantic enrichment improved only slightly the relation classification. Considering that the external resources used to retrieve the semantic information are very limited in terms of languages, and require a massive amount of manual work by linguists and language experts, using this type of information for large processing of data does not seem to be particularly beneficial, considering the costs of creating and maintaining such lexical databases.

5.5 Overall Analysis of the Performance

In the literature review, we have presented the scores of the submission for the SemEval 2018 - Task 7. In this section, we will compare our models with the models published in the literature, as well as discussing their features and architectures.

The best scoring model presented in this research achieved the macro-F1 score of 0.50 and a weighted F1 score of 0.65. Based on the ranking presented in Table 1, our model would have ranked at position 11 in this task, among 18 models (including 17 models published in the literature and our model). Considering that our experiments used a simple deep learning architecture, and that our highest scoring model used a word embedding technique that did not depend on pre-trained corpus, we consider that our model achieved a very good result, since all other models proposed complex architectures with several features. In addition, as discussed in the literature review, no other paper evaluated the impact of each of the used features, so it was not clear which features were able to provide the model with information for them to classify each relation, while our research provided a feature engineering analysis, which has shown that neither the embeddings, nor the syntactic information, nor the semantic enrichment are able to provide the model with sufficient information for a good performance in the relation extraction, given that the classes are not well-defined from the data.

For example, the relations USAGE, MODEL-FEATURE and PART_WHOLE have a great degree of overlap in the classification, as we can see from the confusion matrix in Figure 21.

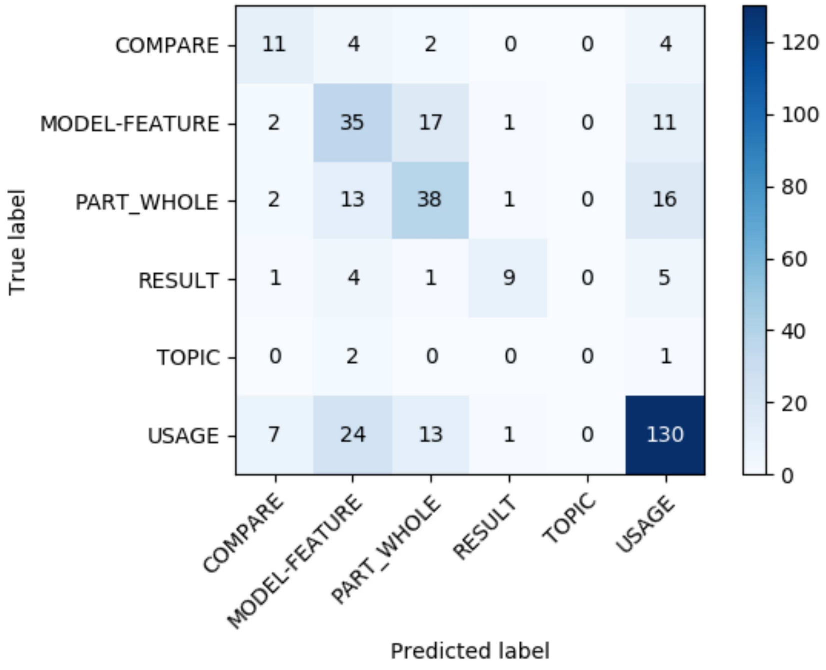


Figure 21 – Confusion Matrix for classifier using Tokenizer word embeddings

We can see that for MODEL-FEATURE, 35 instances were correctly classified, 17 were classified as PART-WHOLE, and 11 as USAGE. For PART_WHOLE, 38 instances were correctly classified, 13 were classified as MODEL-FEATURE, as 16 as USAGE. For USAGE, 130 instances were correctly classified, 24 we classified as MODEL-FEATURE, and 13 as PART_WHOLE.

This degree of overlap between these relation can be seen when we look at some sentences in the corpora that were incorrectly classified. Let us consider sentences that contain the prepo-

Sentence	Predicted Label	True Label
dependency analyser of italian	MODEL-FEATURE	USAGE
morphological analysis of english	MODEL-FEATURE	USAGE
training corpus of spanish	MODEL-FEATURE	PART_WHOLE
segmentation of the input corpora	MODEL-FEATURE	USAGE
parse of indian language	MODEL-FEATURE	USAGE
parser of hindi	MODEL-FEATURE	USAGE
annotation of temporal expression	MODEL-FEATURE	USAGE
corpus of spontaneous spoken dialogue	MODEL-FEATURE	PART_WHOLE
syntactic analysis of new sentence	MODEL-FEATURE	USAGE
morphological analysis of english	MODEL-FEATURE	USAGE
training corpus of spanish	MODEL-FEATURE	PART_WHOLE
relative informativeness of a word	PART_WHOLE	MODEL-FEATURE
hand tagging of text	PART_WHOLE	USAGE
reading difficulty of a text passage	PART_WHOLE	MODEL-FEATURE
syntactic description of a fragment of german	PART_WHOLE	MODEL-FEATURE
performance of the prototype system	RESULT	MODEL-FEATURE

Table 7 – Incorrectly classified instances from test data

sition *of*, displayed in Table 7.

As we can see, the structure <entity> of <entity> is quite common between the different relation classes, therefore ts is quite explainable why the neural network is not able to properly differentiate between these classes. Even for a speaker, these classes may not be as defined for some of these sentences, for example *comparing dependency parser of italian*, and *training corpus of spanish*, they represent different relations, but for both we could transpose the noun to the first element of the sentence (*italian dependency parser* and *spanish training corpus*, making a structure called noun adjunct or attributive noun. However, both sentences accept this transformation, while some do not (for instance *syntactic analysis of new sentence*, which again could not be used by the model to characterize these relations. Overall, these relations were the hardest to classify and impacted the most the model's performance, impact which was clear after analyzing the errors made by the model.

Comparing with the literature, the same techniques for word embeddings and the linguistic information were used by the other models, however, since the data itself is not sufficient to correctly separate the classes, the mechanism by which the other models were able to achieve higher scores is not made clear in the analysis of the respective papers. This gap is what we consider the advantage of our research, because we were able to demonstrate that the data itself cannot differentiate between the different relations, which in turn hinders the model ability to achieve a higher score and generalization.

6 CONCLUSION AND FUTURE STEPS

In this research, we have analyzed how the different word embeddings and linguistic information, synctactic and semantic, affect the performance of a deep learning model on a relation extraction task.

Our main objective was to understand if linguistic features improve the performance of the model in this specific information extraction task, especially considering that using linguistic information usually requires specialized resources, such as lexical databases, which require expert knowledge to be built and depend heavily on a linguistic framework. Besides that, lexical databases may not be available for many languages, which may not be available for the deep learning model.

Overall, our model outperformed several other models published in the literature, even though we used a very simple and standard deep learning architecture. In addition to that, we have performed a deeper analysis on how using different linguistic features may not improve the model as much, and this knowledge is helpful especially when it is necessary to use external resources for semantic enrichment.

Our research demonstrated that semantic information does not necessarily improve significantly the generalization and performance of a deep learning model, and the use of these resources may not be cost effective for the performance improvement they provide to the model. In addition to that, we have explored also the impact of different word embeddings techniques, and showed that using different word embeddings techniques by themselves may not improve the model if the data itself has no clear class distinction for the model to be able to learn and predict them correctly. We consider that this research improves on the existing literature and expands the analysis necessary to understand of deep learning models can be used to learn patterns to perform well in this type of task, as well as being able to generalize this knowledge to other data.

Concluding, as future steps, aggregating other types of semantic enrichment could be explored, as well as enhancing the corpora with additional scientific and not scientific data, so that further information is available for the model, considering that we concluded that only the dataset is not sufficient to provide the model with sufficient data for each relation to be classified. Additionally, performing the same experiments using other relation extraction datasets could help further add to our understanding of whether the learned models can be generalized to different datasets, therefore making it more suitable to be used in productive scenarios. This could bring insights to find the optimal setup for this type of task while providing the necessary analysis to understand if the model has achieved a true generalization for this type of task, which only the score cannot provide by its own.

REFERENCES

ALSENTZER, E. et al. Publicly available clinical bert embeddings. **arXiv preprint arXiv:1904.03323**, [S.l.], 2019.

AUGENSTEIN, I. et al. Semeval 2017 task 10: scienceie-extracting keyphrases and relations from scientific publications. **arXiv preprint arXiv:1704.02853**, [S.l.], 2017.

BAKER, C. F.; FILLMORE, C. J.; LOWE, J. B. The berkeley framenet project. In: COMPUTATIONAL LINGUISTICS-VOLUME 1, 17., 1998. **Proceedings. . .** [S.l.: s.n.], 1998. p. 86–90.

BARIK, B.; SIKDAR, U. K.; GAMBÄCK, B. Ntnu at semeval-2018 task 7: classifier ensembling for semantic relation identification and classification in scientific papers. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings. . .** [S.l.: s.n.], 2018. p. 858–862.

BLOOMFIELD, L. Language. 1933. **New York: Holt**, [S.l.], 1933.

BODEN, M. A guide to recurrent neural networks and backpropagation. **the Dallas project**, [S.l.], 2002.

BROSSEAU-VILLENEUVE, B.; NIE, J.-Y.; KANDO, N. Towards an optimal weighting of context words based on distance. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS, 23., 2010. **Proceedings. . .** [S.l.: s.n.], 2010. p. 107–115.

CHOMSKY, N. Aspects ofthe theory ofsyntax. **Cambridge, MA: MITPress**, [S.l.], 1965.

DENG, L.; LIU, Y. **Deep learning in natural language processing**. [S.l.]: Springer, 2018.

DEVLIN, J. et al. Bert: pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, [S.l.], 2018.

DHYANI, D. Ohiostate at semeval-2018 task 7: exploiting data augmentation for relation classification in scientific papers using piecewise convolutional neural networks. **arXiv preprint arXiv:1802.08949**, [S.l.], 2018.

DRAGONI, M. Neurosent-pdi at semeval-2018 task 7: discovering textual relations with a neural network model. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings. . .** [S.l.: s.n.], 2018. p. 848–852.

ELMAN, J. L. Finding structure in time. **Cognitive science**, [S.l.], v. 14, n. 2, p. 179–211, 1990.

EVERT, S. The statistics of word cooccurrences: word pairs and collocations. **University of Stuttgart**, [S.l.], 2005.

FILLMORE, C. J. et al. Frame semantics and the nature of language. In: NEW YORK ACADEMY OF SCIENCES: CONFERENCE ON THE ORIGIN AND DEVELOPMENT OF LANGUAGE AND SPEECH, 1976. **Annals. . .** [S.l.: s.n.], 1976. v. 280, n. 1, p. 20–32.

GÁBOR, K. et al. Semeval-2018 task 7: semantic relation extraction and classification in scientific papers. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 679–688.

GEE, J. P. Literacy, discourse, and linguistics: introduction. **Journal of education**, [S.l.], v. 171, n. 1, p. 5–17, 1989.

GLUHAK, M. et al. Takelab at semeval-2018 task 7: combining sparse and dense features for relation classification in scientific texts. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 842–847.

GOLDBERG, Y. Neural network methods for natural language processing. **Synthesis Lectures on Human Language Technologies**, [S.l.], v. 10, n. 1, p. 1–309, 2017.

GUO, X. et al. A single attention-based combination of cnn and rnn for relation classification. **IEEE Access**, [S.l.], v. 7, p. 12467–12475, 2019.

HASPELMATH, M.; SIMS, A. **Understanding morphology**. [S.l.]: Routledge, 2013.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, [S.l.], v. 9, n. 8, p. 1735–1780, 1997.

IRMER, M. **Briding inferences in discourse interpretation**. 2010. Tese (Doutorado em Ciência da Computação) — Verlag nicht ermittelbar, 2010.

JIN, D. et al. Mit-medg at semeval-2018 task 7: semantic relation classification via convolution neural network. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 798–804.

JOHNSON, C.; FILLMORE, C. J. The framenet tagset for frame-semantic and syntactic coding of predicate-argument structure. In: MEETING OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 1., 2000. **Anais...** [S.l.: s.n.], 2000.

KEIPER, T. et al. Unima at semeval-2018 task 7: semantic relation extraction and classification from scientific publications. In: UNIMA AT SEMEVAL-2018 TASK 7: SEMANTIC RELATION EXTRACTION AND CLASSIFICATION FROM SCIENTIFIC PUBLICATIONS, 2018. **Anais...** [S.l.: s.n.], 2018.

KIM, Y. Convolutional neural networks for sentence classification. **arXiv preprint arXiv:1408.5882**, [S.l.], 2014.

KUMAR, S. A survey of deep learning methods for relation extraction. **arXiv preprint arXiv:1705.03645**, [S.l.], 2017.

LANGACKER, R. W. Cognitive grammar. In: **Concise history of the language sciences**. [S.l.]: Elsevier, 1995. p. 364–368.

LENA, H. et al. Claire at semeval-2018 task 7-extended version. **arXiv preprint arXiv:1804.05825**, [S.l.], 2018.

LEVIN, B. **English verb classes and alternations**: a preliminary investigation. [S.l.]: University of Chicago press, 1993.

LEVY, O.; GOLDBERG, Y. Dependency-based word embeddings. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (VOLUME 2: SHORT PAPERS), 52., 2014. **Proceedings...** [S.l.: s.n.], 2014. p. 302–308.

LUAN, Y.; OSTENDORF, M.; HAJISHIRZI, H. Scientific relation extraction with selectively incorporated concept embeddings. **arXiv preprint arXiv:1808.08643**, [S.l.], 2018.

MAATEN, L. v. d.; HINTON, G. Visualizing data using t-sne. **Journal of machine learning research**, [S.l.], v. 9, n. Nov, p. 2579–2605, 2008.

MACAVANEY, S. et al. Gu irlab at semeval-2018 task 7: tree-lstms for scientific relation classification. **arXiv preprint arXiv:1804.05408**, [S.l.], 2018.

MAHENDRAN, D.; BRAHMANA, C.; MCINNES, B. Scirel at semeval-2018 task 7: a system for semantic relation extraction and classification. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 853–857.

MARTIN, J. H.; JURAFSKY, D. **Speech and language processing**: an introduction to natural language processing, computational linguistics, and speech recognition. [S.l.]: Pearson/Prentice Hall Upper Saddle River, 2009.

MILLER, G. A. Wordnet: a lexical database for english. **Communications of the ACM**, [S.l.], v. 38, n. 11, p. 39–41, 1995.

NOORALAHZADEH, F.; ØVRELID, L.; LØNNING, J. T. Sirius-ltg-uio at semeval-2018 task 7: convolutional neural networks with shortest dependency paths for semantic relation extraction and classification in scientific papers. **arXiv preprint arXiv:1804.08887**, [S.l.], 2018.

PETERS, M. E. et al. Deep contextualized word representations. **arXiv preprint arXiv:1802.05365**, [S.l.], 2018.

PETRUCK, M. R. Frame semantics. **Handbook of pragmatics**, [S.l.], v. 1, p. 13, 1996.

PRATAP, B. et al. Talla at semeval-2018 task 7: hybrid loss optimization for relation classification using convolutional neural networks. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 863–867.

RENSLOW, T.; NEUMANN, G. Lightrel semeval-2018 task 7: lightweight and fast relation classification. **arXiv preprint arXiv:1804.08426**, [S.l.], 2018.

RIEMER, N. **Introducing semantics**. [S.l.]: Cambridge University Press, 2010.

ROTSZTEJN, J.; HOLLENSTEIN, N.; ZHANG, C. Eth-ds3lab at semeval-2018 task 7: effectively combining recurrent and convolutional neural networks for relation classification and extraction. **arXiv preprint arXiv:1804.02042**, [S.l.], 2018.

SAK, H.; SENIOR, A.; BEAUFAYS, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: FIFTEENTH ANNUAL CONFERENCE OF THE INTERNATIONAL SPEECH COMMUNICATION ASSOCIATION, 2014. **Anais...** [S.l.: s.n.], 2014.

SAUSSURE, F. d. Course in general linguistics (trans. wade baskin). **London: Fontana/Collins**, [S.l.], p. 74, 1916.

SKINNER, B. F. **Verbal behavior**. [S.l.]: New York: Appleton-Century-Crofts, 1957.

SYSOEV, A.; MAYOROV, V. Texterra at semeval-2018 task 7: exploiting syntactic information for relation extraction and classification in scientific papers. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 821–825.

TAI, K. S.; SOCHER, R.; MANNING, C. D. Improved semantic representations from tree-structured long short-term memory networks. **arXiv preprint arXiv:1503.00075**, [S.l.], 2015.

WANG, Y.; CUI, L.; ZHANG, Y. Using dynamic embeddings to improve static embeddings. **arXiv preprint arXiv:1911.02929**, [S.l.], 2019.

WAZLAWICK, R. **Metodologia de pesquisa para ciência da computação**. [S.l.]: Elsevier Brasil, 2017. v. 2.

YIN, W. et al. Comparative study of cnn and rnn for natural language processing. **arXiv preprint arXiv:1702.01923**, [S.l.], 2017.

YIN, Z. et al. Ircms at semeval-2018 task 7: evaluating a basic cnn method and traditional pipeline method for relation classification. In: THE 12TH INTERNATIONAL WORKSHOP ON SEMANTIC EVALUATION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 811–815.