Universidade de Évora - Instituto de Investigação e Formação Avançada

Programa de Doutoramento em Informática

Tese de Doutoramento

Semantic relations between sentences: from lexical to linguistically inspired semantic features and beyond

Pedro Miguel Rocha Pereira Fialho

Orientador(es) | Maria Luísa Torres Ribeiro Marques da Silva Coheur

Paulo Miguel Quaresma

Évora 2023

Universidade de Évora - Instituto de Investigação e Formação Avançada

Programa de Doutoramento em Informática

Tese de Doutoramento

Semantic relations between sentences: from lexical to linguistically inspired semantic features and beyond

Pedro Miguel Rocha Pereira Fialho

Orientador(es) | Maria Luísa Torres Ribeiro Marques da Silva Coheur

Paulo Miguel Quaresma

Évora 2023

A tese de doutoramento foi objeto de apreciação e discussão pública pelo seguinte júri nomeado pelo Diretor do Instituto de Investigação e Formação Avançada:

Presidente	Salvador Abreu (Universidade de Évora)
Vogais	Hugo Ricardo Gonçalo Oliveira (Universidade de Coimbra - Faculdade de Ciência e Tecnologia) Irene Pimenta Rodrigues (Universidade de Évora) Nuno Mamede (Instituto Superior Técnico) Pablo Gamallo (Universidad Santiago de Compostela) Paulo Miguel Quaresma (Universidade de Évora) (Orientador)

Évora 2023

UNIVERSIDADE DE ÉVORA

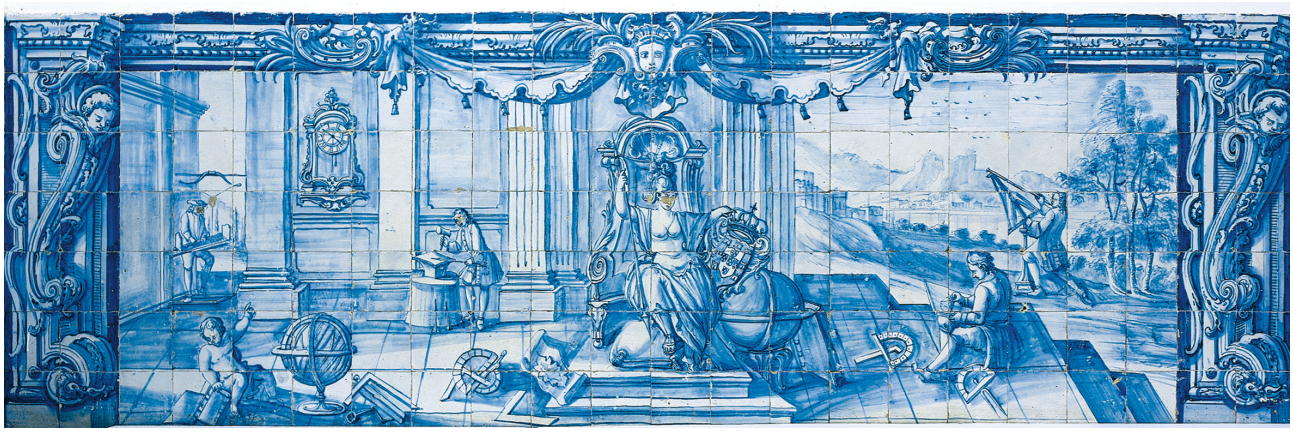
Semantic relations between sentences: from lexical to linguistically inspired semantic features and beyond

Pedro Miguel Rocha Pereira Fialho

Tese apresentada à Universidade de Évora
para obtenção do Grau de Doutor em Informática

Orientador *Paulo Miguel Torres Duarte Quaresma*
Co-orientadora *Maria Luísa Torres Ribeiro Marques da Silva Coheur*

June 27, 2023



INSTITUTO DE INVESTIGAÇÃO E FORMAÇÃO AVANÇADA

UNIVERSIDADE DE ÉVORA

Escola de Ciências e Tecnologia

Departamento de Informática

Semantic relations between sentences: from lexical to linguistically inspired semantic features and beyond

Pedro Miguel Rocha Pereira Fialho

Orientação *Paulo Miguel Torres Duarte Quaresma*
Maria Luísa Torres Ribeiro Marques da Silva Coheur

Informática

Dissertação

June 27, 2023

Aos meus pais.

Agradecimentos

À minha mãe, Maria D'Aires Rocha Pereira Fialho, e ao meu pai, Joaquim Estevão Xarope Fialho.

À minha orientadora, Luísa Coheur, e ao meu orientador, Paulo Quaresma.

Contents

List of Figures

List of Tables

Lista de Acrónimos

AMR Abstract Meaning Representation
ASSIN Avaliação de Similaridade Semântica e Inferência Textual
BERT Bidirectional Encoder Representations from Transformers
BLEU Bilingual Evaluation Understudy
CCG Combinatory Categorical Grammar
DM Double Metaphone
DRS Discourse Representation Structure
DRT Discourse Representation Theory
GLUE General Language Understanding Evaluation
LCS Longest Common Subsequence
LDC Linguistic Data Consortium
LSA Latent Semantic Analysis
MSE Mean Squared Error
MSRP Microsoft Research Paraphrase Corpus
MT Machine Translation
NCD Normalized Compression Distance
NLI Natural Language Inference
NLP Natural Language Processing
NLTK Natural Language Toolkit
NLU Natural Language Understanding
PI Paraphrase Identification
PMB Parallel Meaning Bank
POS Part of Speech
RBF Radial Basis Function
RTE Recognizing Textual Entailment
SemEval The International Workshop on Semantic Evaluation
SICK-BR Brazilian Sentences Involving Compositional Knowledge
SICK Sentences Involving Compositional Knowledge
SNLI Stanford Natural Language Inference
STS Semantic Textual Similarity

SVM Support Vector Machines
TER Translation Edit Rate
TF-IDF Term Frequency–Inverse Document Frequency

Sumário

Relações semânticas entre frases: de aspectos lexicais a aspectos semânticos inspirados em linguística e além destes

Esta tese é dedicada à identificação de equivalência semântica entre frases em língua natural, através do estudo e computação de modelos destinados a tarefas de Processamento de Linguagem Natural relacionadas com alguma forma de equivalência semântica. Em tais tarefas, a partir de duas frases, os nossos modelos produzem uma etiqueta de classificação, que corresponde à relação semântica entre as frases, baseada num conjunto predefinido de possíveis relações semânticas, ou um valor contínuo, que corresponde à similaridade das frases numa escala predefinida. A primeira configuração mencionada corresponde às tarefas de Identificação de Paráfrases e de Inferência em Língua Natural, enquanto que a última configuração mencionada corresponde à tarefa de Similaridade Semântica em Texto.

Apresentamos diversos modelos para Inglês e Português, onde vários tipos de aspectos são considerados, por exemplo baseados em distâncias entre representações alternativas para cada frase, seguindo formalismos semânticos e lexicais, ou vectores contextuais de modelos previamente treinados com Representações Codificadas Bidirecionalmente a partir de Transformadores. Para Inglês, propomos um novo conjunto de aspectos semânticos, a partir da representação formal de semântica em Estruturas de Representação de Discurso. Para Português, os conjuntos de dados apropriados são escassos e não estão disponíveis representações formais de semântica, então implementámos uma avaliação de aspectos actualmente disponíveis, seguindo a configuração de modelos aplicada para Inglês.

Obtivemos resultados competitivos em todas as tarefas, em Inglês e Português, particularmente considerando que os nossos modelos são baseados em ferramentas e tecnologias disponíveis, e que todos os nossos aspectos e modelos são apropriados para computação na maioria dos computadores modernos, excepto os modelos baseados em vectores contextuais. Em particular, para Inglês, os nossos aspectos semânticos a partir de Estruturas de Representação de Discurso melhoram o desempenho de outros modelos, quando integrados no conjunto de aspectos de tais modelos, e obtivemos resultados estado da arte para Português, com modelos baseados em afinação de vectores contextuais para certa tarefa.

Palavras chave: Aspectos lexicais, Estruturas de Representação de Discurso, Identificação de Paráfrases,

Inferência em Língua Natural, Similiaridade Semântica em Texto, Vectores Contextuais

Abstract

Semantic relations between sentences: from lexical to linguistically inspired semantic features and beyond

This thesis is concerned with the identification of semantic equivalence between pairs of natural language sentences, by studying and computing models to address Natural Language Processing tasks where some form of semantic equivalence is assessed. In such tasks, given two sentences, our models output either a class label, corresponding to the semantic relation between the sentences, based on a predefined set of semantic relations, or a continuous score, corresponding to their similarity on a predefined scale. The former setup corresponds to the tasks of Paraphrase Identification and Natural Language Inference, while the latter corresponds to the task of Semantic Textual Similarity.

We present several models for English and Portuguese, where various types of features are considered, for instance based on distances between alternative representations of each sentence, following lexical and semantic frameworks, or embeddings from pre-trained Bidirectional Encoder Representations from Transformers models. For English, a new set of semantic features is proposed, from the formal semantic representation of Discourse Representation Structure. In Portuguese, suitable corpora are scarce and formal semantic representations are unavailable, hence an evaluation of currently available features and corpora is conducted, following the modelling setup employed for English.

Competitive results are achieved on all tasks, for both English and Portuguese, particularly when considering that our models are based on generally available tools and technologies, and that all features and models are suitable for computation in most modern computers, except for those based on embeddings. In particular, for English, our semantic features from DRS are able to improve the performance of other models, when integrated in the feature set of such models, and state of the art results are achieved for Portuguese, with models based on fine tuning embeddings to a specific task.

Keywords: Discourse Representation Structures, Embeddings, Lexical features, Natural Language Inference, Paraphrase Identification, Semantic Textual Similarity

1

Introduction

The ability to identify if two sentences share equivalent semantics is particularly of use to systems that organize data, for instance in clustering of news and user comments, or that accept natural language inputs to be matched against a knowledge base, for instance in question answering and dialogue management tasks. For this thesis we developed models that identify or measure semantic equivalence between a pair of sentences, both for English and Portuguese. Such models are based on lexical features, embeddings based on Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2019] or formal semantic representations, and are computed with both traditional machine learning models, such as Support Vector Machines (SVM), and modern neural networks.

We designed semantic features from Discourse Representation Structure (DRS) [Kamp and Reyle, 1993], a formal semantic representation based on Discourse Representation Theory (DRT), and further combined such features with the remaining types of feature. Our models are evaluated on the tasks of Paraphrase Identification (PI) and Natural Language Inference (NLI), regarding identification of semantic relations, and Semantic Textual Similarity (STS), regarding measurement of semantic equivalence. These tasks are part of the field of Natural Language Processing (NLP), and the resulting models are particularly suitable to address Natural Language Understanding (NLU).

1.1 Motivation

Different types of semantic relations can be found between two sentences. For instance, two sentences are considered to be equivalent (paraphrases) if they share the same meaning. Also, it is possible that one sentence contradicts the other (contradictions), or that no relation at all is found between them (neutral). An entailment relation is also possible, and we say that a source sentence (the premise) entails a target sentence (the conclusion) when the conclusion is most probably true if the source is true [Dagan et al., 2009]. These semantic relations have been widely studied in NLP. For instance, equivalence between sentences is tackled in PI tasks; the detection of relations as contradiction, neutral and entailment is the research target of the NLI task, which is also designated as Recognizing Textual Entailment (RTE); in the STS task the similarity level between two sentences is calculated.

Each one of these tasks can be useful in several NLP scenarios. The task of PI, for instance, can be used for evaluation purposes in Machine Translation: a translation result can be missing a reference, and, still, be a good translation; thus, we should be able to see if it is a paraphrase of some sentence in the reference [Pado et al., 2009]; also, it can be used by a chatbot that has in its knowledge base a set of predefined question/answer pairs: a question submitted by the user needs to be compared with existing questions, and if the user question is a paraphrase of a question in the knowledge base, the system only needs to return the appropriate answer [Fialho et al., 2013, McClendon et al., 2014, Gonçalo Oliveira et al., 2020]; other applications include summarization [Misra et al., 2016], or plagiarism detection [Madnani et al., 2012].

In many cases, just by comparing the shared lexical elements of two sentences (seen as bags of words) we are able to identify their semantic relations. However, in many other cases we need to move to a semantic level. For instance, *Symptoms of influenza include fever and nasal congestion.* and *Fever and nasal congestion are symptoms of influenza.* can be identified as paraphrases by taking advantage of features at a lexical level (for instance, by counting the number of common words). However, the previous sentences and the sentence *A stuffy nose and elevated temperature are signs you may have the flu.*[1] will only be identified as paraphrases if we have access to semantic information, for instance, if we know that *fever* is similar or equal to *elevated temperature* and the same between *nasal congestion* and *stuffy nose*. Thus, a system with the goal of identifying equivalence relations should be able to reason at a semantic level.

Previous work was feature-engineering based. Lexical, syntactic and semantic features extracted from two sentences were combined so that an algorithm such as Support Vector Machines (SVM) could decide the relation between the two sentences [Fialho et al., 2019]. Current work takes advantage of pre-trained models, which can be directly used or tuned to specific domains/tasks. Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2019] is an example of such models. Using BERT, state of the art results were achieved on various NLP tasks, including STS and NLI for Portuguese [Rodrigues et al., 2019a, Cabezudo et al., 2019, Fialho et al., 2020b].

In what concerns semantic features, formal meaning representations can be used as feature providers, as in [Bjerva et al., 2014], which takes advantage of DRS to compute features to address STS and NLI. We take this idea further and propose various features from the DRS obtained from two sentences.

Since DRS are logic based representations, computing features from such representations involved parsing the corresponding logic predicates, to model similarity from components of the predicates. For instance, some of our features rely on counting how many predicates are equivalent, according to a certain similarity metric applied to the components of the predicates. Moreover, we leveraged additional linguistic resources, to further expand such equivalence computation. For instance, other than considering two predicates as equivalent from the exact matching of words in their components, we also rely on linguistic resources to relate words in such components.

[1] `https://examples.yourdictionary.com/examples-of-paraphrasing.html`

Leveraging linguistic resources for word similarity required defining thresholds (from experimentation) that describe a balanced approach to word similarity, such that equivalent words are considered as such, unlike related but not equivalent words. For instance, while using hypernym paths from Wordnet [Fellbaum, 1998] as a linguistic resource, we enforce a limit on how distant a word can be from its original. With such thresholds, we seek that our features are discriminatory.

In addition, we extracted lexical features at no cost (no pre-training) and leveraged semantics from pre-trained BERT models, which are available for several languages, including Portuguese. Since DRS generators only exist for English, BERT models are the only source of semantic features employed in experiments with Portuguese corpora.

1.2 Research Questions

In this thesis, we present a comparative study on the performance of lexical and semantic features in the tasks of PI, NLI and STS. We assume a relatively low resource environment and explore how these features can be used in these tasks for both English and Portuguese. We provide a large set of lexical features, and we focus on the semantics carried by DRS. We also contribute with a set of semantic features extracted from these meaning representations. Traditional feature-based setups are compared and combined with current transfer-learning approaches, based on pre-trained models from BERT. Consequently, our research questions are:

- RQ1: In what extent can lexical and semantic features contribute to the tasks of PI, NLI and STS, both isolated and combined?
- RQ2: How to extract semantic features from DRS, which can contribute to the tasks of PI, NLI and STS for the English language, and how can these be combined with other types of feature?
- RQ3: In our target tasks, what is the performance of pre-trained models for languages other than English, in particular for the Portuguese language, and how is the performance affected when combining lexical features with such models?

We conclude that by combining our lexical and/or semantic features with pre-trained BERT models, we are able to achieve (marginally) better performance than using only BERT. Moreover, the performance of models based only in our lexical and/or semantic features is similar to that of models based on combinations that involve BERT. For English, models based only in semantic features from DRS achieve marginally lower performance than other models, but combining such features with any other type of feature results in better performance than using only the latter. The best overall performance is always achieved with BERT models fine tuned to a specific task, although for some tasks, such as STS for English, the results are similar to those obtained with the contribution of all of our generic and task independent features. In particular, state of the art results are obtained for all the available corpora for Portuguese, in the different tasks.

1.3 Contributions

Our main contribution is a set of semantic features from DRS, which consider aspects that are typically not available in sentence analysis frameworks, such as the scope of negations and implications. Also, some of these features rely on expanding information provided in DRS, based on additional linguistic resources. For instance, some of our features consider synonyms to compute equivalency between words.

Another contribution is the experimentation with various types of features and modelling techniques. For instance, from combining sparse features from embeddings with discrete features from lexical distance metrics, or from computing ensemble models encompassing various forms of machine learning. Moreover, our contributions are applied to languages other than English, namely to Portuguese data for our target tasks.

Several papers were published during this work. We list the ones related with this thesis in the following:

- Fialho, P., Coheur, L., and Quaresma, P. (2020b). Benchmarking natural language inference and semantic textual similarity for portuguese. *Information*, 11(10)

- Fialho, P., Coheur, L., and Quaresma, P. (2020c). To bert or not to bert dealing with possible bert failures in an entailment task. In Lesot, M.-J., Vieira, S., Reformat, M. Z., Carvalho, J. P., Wilbik, A., Bouchon-Meunier, B., and Yager, R. R., editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 734–747, Cham. Springer International Publishing

- Gonçalo Oliveira, H., Ferreira, J., Santos, J., Fialho, P., Rodrigues, R., Coheur, L., and Alves, A. (2020). AIA-BDE: A corpus of FAQs in Portuguese and their variations. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5442–5449, Marseille, France. European Language Resources Association

- Fialho, P., Coheur, L., and Quaresma, P. (2020a). Back to the feature, in entailment detection and similarity measurement for portuguese. In Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., and Gonçalves, T., editors, *Computational Processing of the Portuguese Language*, pages 164–173, Cham. Springer International Publishing

- Fialho, P., Coheur, L., and Quaresma, P. (2019). From Lexical to Semantic Features in Paraphrase Identification. In Rodrigues, R., Janousek, J., Ferreira, L., Coheur, L., Batista, F., and Oliveira, H. G., editors, *8th Symposium on Languages, Applications and Technologies (SLATE 2019)*, volume 74 of *OpenAccess Series in Informatics (OASIcs)*, pages 9:1–9:11, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik

- Fialho, P., Patinho Rodrigues, H., Coheur, L., and Quaresma, P. (2017). L2F/INESC-ID at SemEval-2017 tasks 1 and 2: Lexical and semantic features in word and textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 213–219, Vancouver, Canada. Association for Computational Linguistics

- Fialho, P., Marques, R., Martins, B., Coheur, L., and Quaresma, P. (2016). Inesc-id@assin: Medição de similaridade semântica e reconhecimento de inferência textual. *Linguamática*, 8(2):33–42

- Ameixa, D., Coheur, L., Fialho, P., and Quaresma, P. (2014). Luke, i am your father: Dealing with out-of-domain requests by using movies subtitles. In Bickmore, T., Marsella, S., and Sidner, C., editors, *Intelligent Virtual Agents*, pages 13–21, Cham. Springer International Publishing

- Fialho, P., Coheur, L., Curto, S., Cláudio, P., Costa, Â., Abad, A., Meinedo, H., and Trancoso, I. (2013). Meet EDGAR, a tutoring agent at MONSERRATE. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–66, Sofia, Bulgaria. Association for Computational Linguistics

1.4 Document Overview

This thesis is organized as follows. Chapter 2 provides definitions for the target tasks, an overview of evaluation fora and corpora suitable for such tasks, and popular methods and resources employed in related work. Chapter 3 starts by describing the underlying features of systems developed in this thesis, and concludes with the employed modelling techniques for such features. Chapter 4 presents experiments for the English language, while Chapter 5 presents experiments with Portuguese language, and both describe the experimental setup, such as evaluation metrics and corpora, and include a discussion of particularities of obtained results. Finally, Chapter 6 revisits the research questions, presents conclusions and future work.

2

Related work

In this chapter we define the tasks on which we evaluate the models proposed by this thesis, and some of the approaches followed by other authors to address such tasks. We also present related work based on formal structures and embeddings, since our systems rely on such representations.

2.1 Tasks on Semantic Similarity

We evaluate our systems on the tasks of Paraphrase Identification (PI), Natural Language Inference (NLI) and Semantic Textual Similarity (STS), which are described below. The former two are classification tasks, differing in the set of target labels, while the latter is a regression task. All address the problem of identifying meaning equivalence, and are ultimately targeted at the more general task of NLU.

We describe each of our target tasks in the following, by providing a definition, inherent problems and examples which suggest their application.

2.1.1 Paraphrase Identification

PI is the task of deciding if two sentences are equivalent in meaning. A sentence is a paraphrase of another sentence if the meaning of one is equivalent to the meaning of the other, regardless of their form, length or order as a pair. Meaning equivalence is subjective, and may require domain knowledge not explicit in the sentences, although certain paraphrases are the result of linguistic transformations on a source sentence [y M. Antònia Martí y Horacio Rodríguez, 2010, Bhagat and Hovy, 2013]. For instance, the sentence "The gray clouds were a warning of an approaching storm" is a paraphrase of sentence "The coming storm was foretold by the dark clouds", if one assumes/knows that gray is a dark colour. However, using word "gray" instead of "dark" we obtain sentence "The coming storm was foretold by the gray clouds", which is a also a paraphrase of the former sentences, and does not require domain/world knowledge, regarding the characteristics of colours. Examples of pairs of sentences that are generally considered paraphrases[1] are shown in Figure 2.1, essentially derived from linguistic transformations and requiring only knowledge of natural language usage.

```
The ceiling of the Sistine Chapel was painted by Michelangelo.
Michelangelo painted the Sistine Chapel's ceiling.

It was a spacious room with lit candles all over.
Candles flickered from many areas of the large room.

She was a successful author and speaker.
She found success as a speaker and writer.

The majority of consumers prefer imported cars.
Foreign cars are preferred by most customers.

He has tons of stuff to throw away.
He has to get rid of a lot of junk.

Symptoms of the flu include fever and nasal congestion.
Stuffiness and elevated temperature are signs of the flu.
```

Figure 2.1: Examples of pairs of sentences considered paraphrases.

Paraphrases are symmetric relations, but from a practical perspective we inspect if a target sentence is a paraphrase of a source sentence. Systems aimed at PI classify an input pair of sentences as positive, if equivalent, or negative otherwise.

2.1.2 Natural Language Inference

Paraphrases are a special type of entailment, namely bidirectional entailment. Entailment can be defined as a relationship between two natural language units (e.g., between two sentences) where the truth of one requires the truth of the other. We can say that a sentence A entails a sentence B if and only if whenever A is true, B is also true (and the opposite direction is not implicit).

[1] Obtained from https://www.uobabylon.edu.iq/eprints/eprint_3_24903_1591.doc, https://stranatalantov.com/uploads/publishing/83369_78888.docx and http://www.paraphraseexample.org/one-reasonable-online-paraphrasing-service/examples-of-paraphrasing-sentences-that-work-for-you/

Entailment detection is the main purpose of the RTE task, where, given two sentences, considered as a text and its hypothesis, the task is to decide if it is possible to assume the true in the hypothesis given the text [Dagan et al., 2009]. For instance, from the text sentence "The drugs that slow down or halt Alzheimer's disease work best the earlier you administer them." it is possible to assert the hypothesis sentence "Alzheimer's disease is treated using drugs.", hence the text entails the hypothesis.

The task of RTE was initially defined to label two sentences as entailed or not entailed, and later extended to define entailment as a directional relationship, where a target sentence (the hypothesis) entails a source sentence (the text) but the opposite direction is not required/implicit [Bar-Haim et al., 2006]. Further extensions led to a definition of the task to comprise a distinction among not entailed cases, as either contradictions or neutral (no entailment) [Dagan et al., 2009]. With these latter extensions, RTE is typically named as the task of NLI, to indicate that more categories other than entailment are involved, and that identification of such categories in a open domain setting involves inference capabilities beyond the linguistic definition of entailment [Poliak, 2020].

Examples of pairs of sentences corresponding to the labels entailment, contradiction and neutral, are shown in Figure 2.2, as obtained from the corpus for the PASCAL RTE challenge [Bar-Haim et al., 2006] and the MultiNLI corpus [Williams et al., 2018].

```
TEXT: Yoko Ono unveiled a bronze statue of her late husband, John Lennon, to
complete the official renaming of England's Liverpool Airport as Liverpool John
Lennon Airport.
HYPOTHESIS: Yoko Ono is John Lennon's widow.
LABEL: entailment

TEXT: At the end of Rue des Francs-Bourgeois is what many consider to be the city's
most handsome residential square, the Place des Vosges, with its stone and red
brick facades.
HYPOTHESIS: Place des Vosges is constructed entirely of gray marble.
LABEL: contradiction

TEXT: She smiled back.
HYPOTHESIS: She was so happy she couldn't stop smiling.
LABEL: neutral
```

Figure 2.2: Examples of pairs of sentences considered entailment, contradiction and neutral.

2.1.3 Semantic Textual Similarity

The judgment of being a paraphrase is prone to partial matches, such as when two sentences include the same meaning, but one of them also includes information not present on the other. The task of STS introduces a grading scale for the amount of shared meaning in a pair [Agirre et al., 2012]. Instead of a strict decision, meaning equivalence is measured in a continuous scale, with a minimum for dissimilarity, a maximum for equivalence and various degrees in between according to the amount and relevance of shared semantics/content. As such, STS is a regression task, where the aim is to find a continuous value, usually between 1 and 5, for the similarity among two sentences.

Discrete values within a scale for STS scores are typically defined and assigned to a textual description, such as in the following scale definition, employed in an evaluation for systems that output STS scores

[Fonseca et al., 2016]. Such textual descriptions enable human interpretation of the STS score for a pair of sentences, such that the relatedness between the sentences fits the description in the value that is closest to their score. For each value in the following descriptions, we provide two example pairs of sentences with scores close to the corresponding discrete value, obtained from datasets of the The International Workshop on Semantic Evaluation (SemEval) challenge[2], that include STS scores, where the first example is focused on sentences from factual observations, and the last example is focused on sentences from news headlines. Using discrete values, it is possible to identify the relatedness between two sentences from their continuous STS score, by approximating the score to one of the following values:

1. Completely different sentences, on different subjects:

   ```
   A person is chopping an onion.
   A person is riding an old motorcycle.

   Food price hikes raise concerns in Iran.
   American Chris Horner wins Tour of Spain.
   ```

2. Sentences are not related, but are roughly on the same subject:

   ```
   A man is performing a dance.
   A man is praying.

   Google releases Nexus 5 phone with Kit Kat.
   Google redesigns search results on PCs.
   ```

3. Sentences are somewhat related. They may describe different facts but share some details:

   ```
   A player is throwing the ball.
   A player is running with the ball.

   Tropical Storm Karen targets US Gulf Coast.
   Tropical Storm Karen weakens as it nears U.S. Gulf Coast.
   ```

4. Sentences are strongly related, but some details differ:

   ```
   Someone is slicing a tomato.
   The person is slicing a vegetable.

   Former Pakistan President Pervez Musharraf arrested again.
   Former Pakistan military ruler Pervez Musharraf granted bail
   ```

5. Sentences mean essentially the same thing:

   ```
   A girl in white is dancing.
   A girl is wearing white clothes and is dancing.

   Spain approves new restrictive abortion law.
   Spanish government approves tight restrictions on abortion.
   ```

[2] https://semeval.github.io

Such descriptions are particularly useful for annotators as guidelines in assigning a similarity value for a pair of sentences, such as upon producing corpora suitable to train models that address the STS task [Agirre et al., 2012].

In this thesis we employ datasets based on a 5 value scale, and the previously shown descriptions of such values. Other definitions of the STS task employ a 6 value scale [Agirre et al., 2012], with 0 as the initial value, and where the previously shown value 4 is split in two values, to distinguish important from unimportant details that differ between two sentences. The previously shown examples from news headlines were obtained from datasets of the SemEval challenge[3], based on such scale, and adapted to the 5 value scale, hence the original STS scores are not shown.

2.2 Evaluation Fora

The aforementioned tasks are the target of several evaluation fora, where participants are usually provided with a collection of pairs of sentences annotated with the target output, envisaged for model training, and a collection of pairs of sentences without the target output, envisaged as a test set.

For certain models and tasks, studies suggest that training a model with the train set from such an evaluation fora, and testing that model with the test set from a different evaluation fora, results in a loss of performance, compared to training and testing such model with data from the same evaluation fora [Talman and Chatzikyriakidis, 2019].

The availability of evaluation fora focused on the problem of RTE has fostered the experimentation with a number of data-driven approaches. Specifically, the availability of RTE datasets for supervised training made it possible to formulate the problem as a classification task, where features are extracted from the training examples and then used by machine learning algorithms in order to build a classifier, which is finally applied to the test data to classify each pair of sentences/phrases as either entailed or not.

A benchmark for systems aimed at performing RTE was initially developed in the PASCAL challenge series [Bar-Haim et al., 2014]. Later challenges and corpora mostly followed the task of NLI, usually by labelling pairs of sentences as entailment, contradiction or neutral. Popular evaluation fora on our target tasks were designed on SemEval[4], which has supplied different data and challenges in each of its series, including some of the first corpora aimed at NLI [Marelli et al., 2014a] and STS [Agirre et al., 2012], and a corpora for PI on informal language, based on examples obtained from Twitter messages [Xu et al., 2015].

In SemEval, STS tasks based on multilingual corpora were introduced, namely in cross-lingual [Agirre et al., 2016, Cer et al., 2017] and monolingual [Agirre et al., 2014, Agirre et al., 2015] variants, where the aim of the former is to evaluate STS between examples from different languages, while the latter targets examples within the same language, and includes languages other than English. SemEval also introduced an alternative definition of STS, named interpretable STS [Agirre et al., 2015], where the aim is to align segments between a pair of sentences, and annotate each segment with both an STS score and a label that explains their relation, from a predefined set of labels comprehending relations such as opposition and similarity based on different levels of specificity.

Another benchmark is the General Language Understanding Evaluation (GLUE) [Wang et al., 2018], which was designed to evaluate systems for their joint performance on multiple NLU tasks, where the global score for a participating system is an average of its scores on all tasks, and the score in a task is an average from all the evaluation metrics defined for the task. Some of the GLUE tasks do not follow previous tasks

[3] https://github.com/brmson/dataset-sts/tree/master/data/sts/semeval-sts
[4] https://semeval.github.io

that employ the same corpora. For instance, GLUE comprehends two PI tasks based on different existing corpora, where for one of such tasks is specified a validation partition, undefined in the original corpus, and for the other PI task is employed a new test set, with data unavailable in the original corpus. The evaluation of a system with GLUE requires submitting predictions to a remote platform, which outputs global and task based scores. As such, test sets are not provided, therefore tasks using test data different from the original corpora definition are not reproducible, such as the latter mentioned PI task. Following GLUE, the SuperGLUE benchmark was introduced [Wang et al., 2019a], comprehending a different combination of challenges, more recent corpora, and some new tasks, although the RTE task is the same as in GLUE.

To the best of our knowledge, the first NLP shared task focused on similarity for Portuguese sentences was Avaliação de Similaridade Semântica e Inferência Textual (ASSIN) [Fonseca et al., 2016], where the aim is to identify if a pair of sentences is a paraphrase, an entailment case, or none of these, using examples from news sources in Brazilian and European Portuguese subsets[5]. We developed a system to address such task, and obtained one of the best results. The ASSIN2 shared task [Real et al., 2020] followed ASSIN, but instead aimed at identifying pairs of Brazilian Portuguese sentences as either entailment or neutral [Real et al., 2020].

2.3 Corpora

Participants of evaluation fora address a particular configuration of the tasks, relative to corpora and language. Corpora is employed to achieve a model of the task, trained on a collection of examples labelled according to the expected output of a certain task. Various benchmarks exist, to evaluate a certain approach on various tasks and according to popular corpora, such as SentEval [Conneau and Kiela, 2018], GLUE [Wang et al., 2018], or its follow up SuperGLUE [Wang et al., 2019a]. Some of the corpora presented in the following result from previously mentioned evaluation fora.

For PI, supervised models are based on corpora composed by pairs of sentences labeled as true or false (1 or 0, for instance) considering that they are or they are not paraphrases, respectively. The MSRP [Dolan and Brockett, 2005] is one of such corpus, for which early results from various systems are published in a ranking[6]. An annotated version of MSRP, relative to linguistic phenomena, is also available [Kovatchev et al., 2018]. Other corpora for PI are, for instance, based on Twitter messages [Lan et al., 2017], open domain questions from Quora [Zhiguo Wang, 2017], or books [Potthast et al., 2010]. Multilingual corpora for PI are for instance based on subtitles [Creutz, 2018] or language learning resources [Scherrer, 2020].

A recent line of work is to explore model robustness by generating adversarial examples that lower the performance of an otherwise competitive model. Corpora composed of adversarial examples were designed to explore model robustness in PI, both for English [Zhang et al., 2019b] and multilingual examples [Yang et al., 2019a], although none is available for Portuguese, to the best of our knowledge. For NLI, adversarial examples were employed to improve the robustness of a model to corpora composed of adversarial examples [Minervini and Riedel, 2018].

NLI and STS are represented in the SICK corpus [Marelli et al., 2014b], composed by 10000 pairs of sentences seeded from corpora of image and video captions, which are expanded by rule based transformations to introduce particular linguistic phenomena, such as negations. SICK is annotated by crowd-sourcing, and was the target of a shared task on one of the SemEval series [Marelli et al., 2014a].

Each instance in SICK, that is, each pair of sentences, is labelled as entailment, contradiction or neutral regarding the semantic relation between the two sentences. For instance, the pair composed by the sentences

[5] http://propor2016.di.fc.ul.pt/?page_id=381

[6] https://aclweb.org/aclwiki/Paraphrase_Identification_(State_of_the_art)

"Three kids are jumping in the leaves" and "Three boys are jumping in the leaves", is labeled as entailment, while the former sentence paired with "Three kids are sitting in the leaves" is labeled as neutral. An example of a pair labeled as contradiction in SICK is composed by the sentences "Nobody is riding the bicycle on one wheel" and "A person is riding the bicycle on one wheel".

Following SICK, the much larger Stanford Natural Language Inference (SNLI) corpus [Bowman et al., 2015] was released, containing 570000 examples also seeded from corpora of captions and annotated by crowd-sourcing, but instead expanded by crowd-sourcing. SNLI inspired the creation of other corpora on NLI, for instance the e-SNLI corpus [Camburu et al., 2018a], that augments SNLI with natural language explanations for the annotations, or the MultiNLI corpus [Williams et al., 2018], that follows the same design procedure and size of SNLI, but instead of captions includes sentence pairs from other text genres and sources, such as fiction books or transcripts of conversations. MultiNLI is one of the targets of the GLUE benchmark [Wang et al., 2018].

Other corpora for NLI include the CommitmentBank corpus [Jiang and de Marneffe, 2019], which is a recast to NLI of a corpus representing varying degrees of belief on clauses (and part of the SuperGLUE benchmark [Wang et al., 2019a]), the MedNLI corpus, based on clinical notes [Romanov and Shivade, 2018], and the Dialogue NLI corpus, based on crowd-sourced dialogues driven from artificial personas/profiles [Welleck et al., 2019]. As with the CommitmentBank corpus, a recast was also applied to corpora for other semantic phenomena, thus increasing the diversity of available examples in the NLI format [Poliak et al., 2018].

As modern machine learning architectures particularly leverage large data collections, recent approaches suitable for NLI are mostly applied to corpora such as SNLI or MultiNLI, both for their greater size and complexity. One of such approaches is the BERT model [Devlin et al., 2019], which achieves competitive results on various NLU tasks, as shown from its performance on the GLUE benchmark [Devlin et al., 2019], but also specifically in NLI, such as when applied only to MultiNLI [Devlin et al., 2019] or to SNLI [Zhang et al., 2019c].

To the best of our knowledge, the first corpus to include entailment labels and similarity values for Portuguese sentences was the ASSIN corpus [Fonseca et al., 2016], produced in the previously mentioned shared task with the same name. ASSIN contains pairs of sentences from news sources, split into subsets for Brazilian and European Portuguese. Recently, a translation of SICK sentences to Portuguese, the SICK-BR corpus [Real et al., 2018], was made available. The ASSIN2 corpus is based on SICK-BR entailment and neutral examples, expanded by lexical transformations [Real et al., 2020]. In this thesis we evaluate our models on all of these Portuguese corpora.

2.4 State of the Art

Most approaches to our target tasks use machine learning algorithms (e.g., linear classifiers) with a variety of features, including lexical, syntactic and semantic features. The most advanced approaches are based on semantic analysis, using various forms of semantics, from formal and vector representations.

In the following, we first describe popular approaches to each of the tasks involved in this thesis. Some works employ the same approach to multiple tasks [Wu, 2005, Beltagy et al., 2014, Zhao et al., 2014, Filice et al., 2015, He et al., 2015, Yin et al., 2016, Conneau et al., 2017, Zhiguo Wang, 2017, Xiong et al., 2021], while others study the performance of various approaches in multiple tasks [Xu et al., 2015, Lan and Xu, 2018] or survey existing methods for a certain task [Dagan et al., 2009, Androutsopoulos and Malakasiotis, 2010, Chandrasekaran and Mago, 2021]. At last, we present a review of approaches that employ some form of semantic analysis, including those that we address in our experiments.

2.4.1 Paraphrase Identification

As previously mentioned, PI is the task of labelling a pair of sentences relative to whether the sentences are equivalent in meaning, or not. Applications of PI include retrieving the appropriate answer to a question [McClendon et al., 2014], or clustering dialogues [Misra et al., 2016].

Different approaches to PI have been produced along the years with some sort of combination of lexical, syntactic and semantic features. A simple approach is the bag-of-words strategy, in which the comparison of sentences in a given input pair is calculated using a cosine similarity score. If the score is greater than a threshold value (determined manually or learned from data), the sentences are classified as paraphrases [Mihalcea et al., 2006, Fernando and Stevenson, 2008].

As the complexity of language in input sentences may not fit generic methods to measure equivalence, an approach is to convert the sentences into a canonical form, for instance through a set of rules. The definition for canonical form varies. For instance, in [Zhang and Patrick, 2005] the canonical form of a sentence is obtained by changing from passive to active voice, and PI is assessed from such simplified sentencets, leveraging decision trees and lexical matching features such as the edit distance between the tokens. A different example of canonical form can be found in [Brun and Hagège, 2003], where PI is assessed from domain specific predicates, obtained from symbolic methods and dependency parsing, which represent the original sentences.

In addition to features based on lexical matching, some authors proposed classification approaches using a combination of lexical and semantic features and heuristics (e.g., negation patterns [Ul-Qayyum and Wasif, 2012]). One of such semantic features is based on comparing words of a certain Part of Speech (POS), such as nouns, verbs or adjectives, by leveraging their antonyms and synonyms [Ul-Qayyum and Wasif, 2012]. Additional semantic features are, for instance, based on interpreting expressions involving numbers, and from comparing proper nouns [Kozareva and Montoyo, 2006]. Beyond features expressing semantic aspects, some works compute pragmatic features for PI, which, for instance, leverage language polarity aspects, or the semantics of negations [Dey et al., 2016]. Some of the features we compute in this thesis also consider antonyms/synonyms, negations and expressions involving numbers.

Other than to directly compute features, semantics are also employed in PI to form alternative representations of sentences, from which features are computed. One of the systems using such approach leverages predicate-argument structures of verbs and their arguments, derived from semantic role labeling and syntactic analysis [Qiu et al., 2006]. Features are computed from such structures, and one of the strategies to decide if a given pair is a paraphrase involves assessing the coherence and significance of the dissimilar parts between sentences, relative to the similar parts. Later works also explored the relevance of parts of a sentence to the paraphrase relation [Filice and Moschitti, 2016]. In this thesis we also leverage alternative representations of sentences, namely DRS, a formal semantic representation that also includes predicate-argument structures derived from Combinatory Categorical Grammar (CCG) parsing [Bos, 2008].

As paraphrases regularly involve synonyms or other forms of word relatedness, some approaches rely on word-level similarity methods to determine if a sentence is a paraphrase of another sentence [Mihalcea et al., 2006, Fernando and Stevenson, 2008]. These methods are typically based in word-to-word similarity measures, which include knowledge and corpus-based measures. With knowledge-based measures, similarity is derived from linguistic resources that specify properties and relations for words, such as their synonyms and hierarchies, as available in the WordNet database [Fellbaum, 1998]. In corpus-based measures, each word corresponds to a fixed sized vector, and similarity is computed from such vectors, for instance as a result of their cosine distance. Distributional models are employed on text collections to compute latent features that compose the vectors, for instance based on the co-occurrence of words, such as defined in the Latent Semantic Analysis (LSA) model [Landauer and Dumais, 1997].

One of the approaches for PI using word-to-word similarity is based on multiple measures, both knowledge and corpus-based, but is primarily focused on the most similar words [Mihalcea et al., 2006]. Another approach instead considers the similarity of all words, but only employs knowledge-based methods [Fernando and Stevenson, 2008].

Modern word-to-word similarity is based on vectors, as in the previously mentioned corpus-based measures, but instead using context-based distributional models [Bengio et al., 2003]. Vectors produced with such models are named word embeddings, and contain latent features based on the surroundings of a word in its various usages within a collection of texts. For instance, with the Skip-gram model of the word2vec framework [Mikolov et al., 2013], the embedding of a word is computed relative to its neighbour words, such that two words frequently surrounded by the same neighbour words are considered similar. Word embeddings are typically employed from pre-trained models available with each framework, and various frameworks exist, such as FastText [Bojanowski et al., 2017] and ELMo [Peters et al., 2018]. In this thesis, we employ FastText to compute some of our features.

A popular method to obtain the vector of a sentence from the embeddings of its words is by summing the embeddings, since some arithmetic operations on embeddings correspond to an interpretation in natural language. For instance, with one of the pre-trained models of word2vec, subtracting the vector for word "Spain" from the vector for "Madrid", and adding the vector for "France", results in a vector close to that of the word "Paris" [Mikolov et al., 2013]. However, some models produce word embeddings aware of sentence structure, for instance leveraging dependency parsing [Levy and Goldberg, 2014].

One of the works using word embeddings for PI computes the vector for a sentence as a sum of the embeddings of its words and phrases, using WordNet to define phrases and word2vec to compute the embeddings [Yin and Schütze, 2015]. Another approach also employs word2vec, but instead relies on pre-trained models and represents each sentence with the embeddings arranged in a matrix [Wang et al., 2016]. In both systems, similarity is modelled from which contents are shared by both sentences or missing in one of them, which in the former system results in weighted embeddings, and in the latter system results in new vectors, derived by manipulation of the embeddings according to such decomposition in similar and dissimilar parts. A related approach also represents sentences as a matrix of pre-trained word embeddings from word2vec, and is also based on manipulating the embeddings, but the dependencies between sentences are driven by techniques from modern neural networks, which for instance enable to consider phrases of various lengths without using external thesaurus such as WordNet [Yin et al., 2016].

Although word2vec is a popular framework to compute word embeddings, some works employ other frameworks to address PI. One of such works combines pre-trained word embeddings from various frameworks, and also embeddings for POS, trained with word2vec [He et al., 2015]. This system relies on modern neural networks to compute a vector representation for each sentence based on various embeddings, and model similarity between such representations in a structured manner, by employing vector similarity metrics, such as cosine distance, on particular regions of the vectors. Another work, also evaluated on PI, instead designs a new framework to compute word embeddings, that considers multiple word senses, and enforces syntactic and semantic constraints by training with negative examples [Cheng and Kartsaklis, 2015]. Such work implements a compositional model [Sadrzadeh and Kartsaklis, 2016], which allows to intrinsically scale embedding computation to arbitrary spans of text, such as phrases and sentences, and to train embeddings jointly with other layers of parameters, such as those modelling syntactic and semantic information.

Currently, one of the most popular methods to compute embeddings is the BERT model. Relying on modern neural networks, it is also based on context, as traditional word embeddings, but instead allows to obtain dynamic word embeddings which vary with the input sentence [Devlin et al., 2019]. Also unlike traditional word embeddings, the BERT model enables to adapt generic embeddings to a certain task, by fine tuning the model. The BERT model achieves state of the art results on various tasks, including

PI, and is the underlying model in various state of the art systems. However, the full BERT model is computationally demanding, for which some works propose smaller BERT models for PI [Arase and Tsujii, 2021]. We leverage the BERT model for some of our experiments, hence more details of the model are provided in the next chapter.

Although BERT achieves state of the art performance on curated data, it is prone to errors when such inputs are modified, for instance by replacing certain words. Some works study robustness issues in modern neural models by leveraging such adversarial examples. One of such works leverages BERT to find replacement words that reduce the performance of various models on PI, while providing evidence that model robustness is improved when adversarial examples produced with such method are included in training [Shi and Huang, 2020].

For PI, the BERT model was employed in various works, for instance by fine tuning the generic model [Devlin et al., 2019]. Another approach is to combine BERT embeddings with additional information, for instance with embeddings of complementary text from a knowledge base [Wang et al., 2021], or with vector representations of syntax and semantics, from neural models for dependency parsing and semantic role labelling, respectively [Liu et al., 2020].

Before the BERT model, the combination of vectors and features from different levels of linguistic analysis was already employed for PI. One of the systems that followed such approach represents the pair of sentences as a single vector, to which additional lexical and syntactic features are appended, such as from counting unigram overlap or from distance metrics between dependency parse trees [Ji and Eisenstein, 2013]. The addition of features to the sentence vectors is what enabled this system to achieve one of the best scores in the MSRP corpus, although these vectors were not derived from context-based models. In this thesis we also combine vectors and discrete features from different levels of linguistic analysis.

Another form of leveraging vectors for PI is in composing alternative representations of sentences. One of the systems leveraging such approach composes predicate-argument structures as a graph, where embeddings and lexical information, such as POS, are attached to the graph nodes [Liang et al., 2016]. A graph is computed for each sentence, and ultimately a feature vector is computed from the pair of graphs, based on an alignment between the graphs, which considers the different types of information encoded in the graph. A similar approach employs graph kernels to model the structural similarities between graphs, although node similarity is not employed [Filice et al., 2015].

Methods based in alignments, such as in metrics inspired in summarization and Machine Translation (MT) evaluation, are commonly used for PI. One of the systems using such approach relies on string alignment metrics from the field of MT [Madnani et al., 2012]. Although the use of MT metrics for PI is not novel [Finch et al., 2005], the authors of such system merit from a thorough re-assessment of these metrics conjointly with the creation of new metrics, and achieved one of the best results on the MSRP data.

Popular features employed in PI were primarily designed for MT evaluation, such as Bilingual Evaluation Understudy (BLEU) [Papineni et al., 2002a]. These and many other features have been applied to PI, and there are toolkits that allow to extract features from different linguistic levels. For instance, HARRY [Rieck and Wressnegger, 2016] provides lexical features from string similarity metrics applied to various forms of decomposing words, and SEMILAR [Rus et al., 2013] provides sentence-to-sentence and word-to-word similarity metrics, from knowledge and corpus-based methods.

Regarding syntactic representations, some works take advantage of these structures on PI. Many of such approaches rely on models derived from structured representations, for instance probabilistic models based on approximate alignments between parse trees [Das and Smith, 2009], kernel based models to compute features based on which sub trees are the same structure wise [Filice et al., 2015, Fialho et al., 2019], or neural networks to adjust word vectors based on the structure of parse trees [Socher et al., 2011]. Other

forms include computing a single score based on comparison of syntactic structures [Rus et al., 2008], or engineering a set of features for the overlap between structures [Wan et al., 2006]. As the nodes of syntactic structures contain words from the source sentence, some works increase the matching possibilities between structures by leveraging synonyms from a thesaurus such as WordNet [Rus et al., 2008] or word vectors [Filice et al., 2015]. Most of these approaches also leverage lexical information to compute additional features, which are combined with the syntactic model.

In the semantic features domain, formal meaning representations of sentences are also employed for PI, for instance by combining features from different levels of language analysis with scores relative to the overlap among DRS [van der Goot and van Noord, 2015a], or Abstract Meaning Representation (AMR) [Issa et al., 2018, Fialho et al., 2019], which is another formal semantic representation.

Some works evaluate PI approaches in corpora for languages other than English, for instance in Portuguese [Anchiêta et al., 2020] or in multiple languages available from subtitles [Sjöblom et al., 2018]. In this thesis we also evaluate our PI methods in Portuguese corpora.

2.4.2 Natural Language Inference

As previously mentioned, NLI is the task of classifying two sentences relative to their semantic relation, according to a predefined set of possible semantic relations, such as contradiction and entailment. A similar task is RTE, where the semantic relation for a pair of sentences is either entailment or not entailment. Although RTE is a particular configuration of NLI, where the set of possible semantic relations contains only the two mentioned options, in the following we distinguish approaches for each task. Applications of these tasks include assessing the quality of MT outputs [Pado et al., 2009], the consistency of dialogues [Welleck et al., 2019] or producing captions for images [Shi et al., 2021].

For the RTE task, various works consider a lexical and syntactic approach. One of such works was described in the previous section, as it is also evaluated in PI, and is based on combining Tree Kernels [Moschitti, 2006], to model the structured information in parse trees, and lexical features, to leverage word level similarities [Filice et al., 2015]. Another work describes a composition of several modules, namely a pre-processing module, to replace or remove particular characters, which some similarity metrics are not prepared to process (such as accents), a lexical similarity module, which computes features based on WordNet and lexical distances, and a syntactic similarity module, based on comparing elements from the dependency analysis of each sentence, such as nouns and numbers [Pakray et al., 2011]. A similar system further includes corpus-based measures, and also employs a pre-processing module, which for instance replaces words by their lemmas, and expands contractions, such as to transform “doesn’t” into “does not” [Zhao et al., 2014].

Another system addressing the RTE task uses machine learning methods with features based on lexical information and predicate-argument structures [Tsuchida and Ishikawa, 2011]. The underlying idea is to identify the text-hypothesis pairs that have a high entailment score but are in fact not entailed, such that false-positive pairs classified by the system’s lexical-level module can later be rejected by the sentence-level module.

In addressing RTE with semantic features, some works leverage DRS to represent sentences, and compute features based on the overlap between components of the DRS for each sentence. An early system with such approach converts the DRS to first-order logic, to enable usage of automatic reasoning tools, such as theorem proving [Bos and Markert, 2005]. A later system is based on combining the former with features based on other semantic resources, such as the WordNet thesaurus and word2vec embeddings, and also with lexical features, for instance based on word overlap [Bjerva et al., 2014]. Another system instead

leverages a combination of DRS and Markov models [Beltagy et al., 2014].

Some systems leverage semantic representations other than DRS, although their method for RTE is also based on representing sentences as logical formulas, and deriving the entailment judgment by theorem proving [Abzianidze, 2015, Martínez-Gómez et al., 2017]. The frameworks employed by such systems share similar properties with DRS and are also based on CCG parsing.

Most recent systems address the NLI task, and leverage embeddings based on BERT models. One of such works addresses NLI as a base task to train a generic model, which is then employed to address multiple domains and NLP tasks, by leveraging a few examples from the target domain, eventually converted to the NLI [Yin et al., 2020]. Similarly, but earlier than the BERT model, another work designed sentence embeddings suitable for transfer learning, such that the features learned from training a model for NLI are employed to address other NLP tasks [Conneau et al., 2017].

Earlier than the BERT model, competitive results were obtained in NLI with system based on neural networks and word embeddings, using technologies later leveraged by BERT models, such as bidirectional word representations and attention models [Chen et al., 2017, Zhiguo Wang, 2017]. One of such systems is further complemented by leveraging syntactic information from parse trees, within the neural network [Chen et al., 2017].

As with PI, the BERT model was employed for NLI in various works, for instance by fine tuning the generic model [Devlin et al., 2019, Jiang and de Marneffe, 2019]. Other approaches include training a BERT model for an augmented train set of a given NLI corpus, obtained by generating complementary sentences that support the NLI label of an example [Chen et al., 2021a]. Such system produces a form of natural language explanations for its decisions, relative to NLI, while other works have studied different forms of explanation [Jiang et al., 2021].

A recent work relies on NLI to suggest that, despite the state of the art performance of BERT models, language knowledge is not embedded in the corresponding vectors, since the performance of such models in NLI is similar regardless of the grammatical correctness on the input sentences [Sinha et al., 2021]. Other works mention that the performance of models based on corpora typically employed in benchmark frameworks may not generalize to non-curated examples, due to problems introduced by corpora construction techniques [Herlihy and Rudinger, 2021].

Some works focus specifically on BERT failures in NLI, such as to hypothesize that the success of BERT relies on the occurrence of certain linguistic patterns in the data [McCoy et al., 2019], to suggest that BERT does not implicitly learn linguistic priors and is mostly driven by statistical phenomena [Jiang and de Marneffe, 2019], or to employ an alternative model to address inputs where it is predicted that BERT will fail [Fialho et al., 2020c]. In NLI models, performance is further evaluated with challenging examples, relative to linguistic phenomena [Naik et al., 2018], or adversarial examples, for better model robustness [Minervini and Riedel, 2018].

Other works combine BERT models with symbolic approaches, for instance based on polarity marks [Chen et al., 2021b]. In this thesis, we design features from DRS, which is also a symbolic representation, and combine such features with BERT embeddings.

Due to the availability of corpora for NLI in languages other than English, and of multilingual pre-trained models, such as provided with BERT, various works study the performance of popular models, known for state of the art results in English, in other languages. For instance, some works employ monolingual and multilingual BERT models to address NLI in Portuguese, including our own work [Fialho et al., 2020b]. In this thesis we also employ such models and corpora to address NLI in Portuguese.

2.4.3 Semantic Textual Similarity

As some of the available corpora for NLI also provide STS scores, some of the systems addressing NLI also address STS. One of such systems was previously mentioned for NLI, and leverages lexical, syntactic and corpus-based measures, after a pre-processing module to normalize text [Zhao et al., 2014]. Some of these systems also leverage semantics in the form of DRS, based on automatic reasoning tools [Bjerva et al., 2014] or models combining logic and statistics [Beltagy et al., 2014].

STS models are also evaluated in PI datasets, using a threshold to transform the semantic similarity score into a binary label [Guo and Diab, 2012].

Regarding syntactic features, some works take advantage of these structures on STS, while leveraging neural networks [Tai et al., 2015]. Other works instead rely only in word-level aspects. One of such works is based on word embeddings, and leverages various types of neural networks to model correspondence between words [He and Lin, 2016]. Another work employs other word-level resources, such as WordNet, and composes the STS score of a pair of sentences from the similarity scores of their components, by designing a model based on alignments between chunks of each sentence [Li and Srikumar, 2016].

Monolingual and multilingual BERT models are also employed to address STS, according to corpora availability, for instance in Portuguese [Fialho et al., 2020b].

2.4.4 Semantic Representations

In this thesis, similarity is computed from the meaning of the full sentence, as obtained from a formal semantic representation, where semantic phenomena occurring in the sentence are enclosed in a structured representation and described with a symbol based notation. Lexical aspects and syntax are subsumed in such semantic representation.

In recent years, the main formal semantic representations are either AMR [Banarescu et al., 2013] or DRS [Kamp and Reyle, 1993]. For tasks on semantic similarity, various recent works leveraged AMR [May, 2016, May and Priyadarshi, 2017], while DRS are less common [Abzianidze et al., 2019]. Early works also leveraged both DRS [Bos, 2015] and AMR [Flanigan et al., 2014].

Modern parsing of formal semantic representations is based on neural networks, such as sequence to sequence models, that rely on supervision from corpora with reliable instances of such representations. For instance, neural AMR parsing based on the Linguistic Data Consortium (LDC) corpus [Zhang et al., 2019a], or neural DRS parsing [van Noord et al., 2019] based on the Parallel Meaning Bank (PMB) corpus [Abzianidze et al., 2017].

While AMR are inherently represented as a graph, DRS are produced in various formats, which include graphs [Basile, 2015] but also a human readable notation made of referents and conditions arranged in boxes, as originally employed in the underlying formulation from DRT [Kamp and Reyle, 1993]. Neural DRS parsing is mostly targeted to produce machine readable DRS, using the clausal notation adopted in the PMB corpus, and while obtaining state of the art results on such corpus, do not provide DRS in a human readable format. The early DRS parser Boxer [Bos, 2008], based on handcrafted language resources and rule based CCG derivations, is able to produce DRS in a human readable boxed format, although its performance is approximately 10% worse than that of neural approaches [van Noord et al., 2019].

This thesis explores DRS, for its greater variety of covered semantic phenomena [Bos, 2016, van Noord et al., 2018a], and relies on the boxed format from the Boxer parser, for providing greater readability. Such format is the target representation for engineering features relative to sentence (and, consequently, DRS)

similarity.

Other than formal semantic representations, embeddings also represent semantics, from learned patterns in collections of text. One of the most popular embedding frameworks is the BERT model. The BERT model [Devlin et al., 2019] achieved state of the art results on various NLP tasks for English, as those in the GLUE benchmark [Wang et al., 2018]. BERT produces contextual and dynamic embeddings from a deep learning architecture based on bidirectional Transformers [Vaswani et al., 2017], such that the embedding of a word is specific to the context in which it is employed, and the same word employed in different context results in different embeddings. Training a BERT model is expensive on time and resources, but pre-trained models (in base or larger versions), based on Wikipedia, were made available on various languages including Portuguese [Pires et al., 2019].

2.5 Summary

In this chapter we provided definitions for the target tasks, an overview of evaluation fora and corpora suitable for such tasks, and popular methods and resources employed in related work.

In the following chapter we start by describing the underlying features of systems developed in this thesis, and conclude with the employed modelling techniques for such features.

3

Back to the feature

We gathered sets of features from different linguistic levels, targeted to represent similarity aspects between two sentences that are input to our models. As such, an input pair of sentences is transformed into a set of features/numbers, suitable to build a model.

In the following we describe our features, and the types of model we employ. First, in Section 3.1 we describe our lexical features, which address similarities based on word and character-level. In Section 3.2 we describe our semantic features, based on the formal semantics of DRS, while also introducing the components of DRS that we employed in designing such features. In the remaining sections we describe how we leverage embeddings for sentence similarity, and the details and architecture of our models, both for feature sets and embeddings.

3.1 Lexical Features

In this thesis, we call *lexical features* to the ones based on different distance metrics calculated between the lexical elements of a sentence, and assuming that these distances can be computed both at the character

or word level. We also assume that words can be transformed in their lexical variants, by applying, for instance, stemming or encoding text into the way it sounds. An example of a lexical feature suitable for words and character representations is the ratio between the lengths of two sentences, obtained by dividing the length of the shorter sentence by the length of the longer. The output of this and other lexical features is shown in Figure 3.1, on the original representation considered for an input pair of sentences, and on some of the alternative representations .

```
Original input:
    Sentence 1: A player is throwing the ball
    Sentence 2: Two teams are competing in a football match

Lexical features:
    Length ratio: 0.75
    Max length: 8
    Min length: 6
--------------------------------
Stemmed representation:
    Sentence 1: a player is throw the ball
    Sentence 2: two team are compet in a footbal match

Lexical features:
    Length ratio: 0.75
    Max length: 8
    Min length: 6
--------------------------------
Representation as character trigrams:
    Sentence 1: a pla lay aye yer is thr hro row owi win ing the bal all
    Sentence 2: two tea eam ams are com omp mpe pet eti tin ing in a foo
                oot otb tba bal all mat atc tch

Lexical features:
    Length ratio: 0.65
    Max length: 23
    Min length: 15
```

Figure 3.1: Example outputs of lexical features, for an example from the SICK corpus. In this example, both the original input and its transformation after stemming produce the same values for the shown features.

Some of the models created in this thesis are based on multiple lexical features. Previous studies, within the area of NLP and also in other fields, have already used similar methods for combining multiple similarity metrics in the context of accessing the similarity between objects [Martins, 2011, Madnani et al., 2012].

The lexical features employed in this thesis are based on the set of features first introduced by [Marques, 2015], and correspond to an updated version of such features, after revisions, which in some cases required employing more modern tools and resources. These were also continuously improved upon participation in the ASSIN [Fonseca et al., 2016] and ASSIN2 [Real et al., 2020] challenges, previously described in Section 2.2, and we achieved state-of-the-art results on the European Portuguese track of ASSIN with the INESC-ID@ASSIN system [Fialho et al., 2016], based on the mentioned lexical features.

Our version of the lexical features from [Marques, 2015] are detailed in the following sections, providing

complementary details to the descriptions in [Marques, 2015], and mentioning where our features differ from the original. To support the description of each feature, the two input sentences may also appear mentioned as text and hypothesis, to consider their order in the pair, or sequences, to consider lexical transformations of the original sentences.

String Similarity The string similarity features considered are:

1. **Longest Common Subsequence**. Considering two sequences of characters, such as from two sentences, the Longest Common Subsequence (LCS) [Cormen et al., 2009] is the longest sequence of characters, from left to right and eventually non consecutive, that occurs in both sequences. This feature is the result of dividing the length of the LCS by the length of the longest sequence in the pair, in order to obtain a value between 0 and 1.

   The original computation for the LCS was replaced by another framework[1].

2. **Edit Distance**. This feature corresponds to the minimum Levenshtein distance [Levenshtein, 1966] required to transform one sentence into the other, considering insertion, deletion and substitution of characters.

3. **Length**. Leveraging the lengths of the two input sentences, 3 features are defined. The first outputs a value between 0 and 1, obtained by dividing the length of the shortest sentence by the length of the longest (length ratio). The other two correspond to the lengths of the shortest and longest sentences.

4. **Cosine Similarity**. The cosine of the angle between two vectors is a popular form to assess their similarity. Given two numeric vectors *a* and *b*, such as for the features of two sentences, their cosine similarity is obtained with a dot product between the two vectors, normalized by vector length, as shown in Equation 3.1. This metric is introduced in NLP textbooks [Jurafsky and Martin, 2009].

$$\text{cosine}(a, b) = \frac{a \cdot b}{||a|| \times ||b||} \tag{3.1}$$

   We transform the text and the hypothesis into vectors based on the number of occurrences of each word (the term frequency representation), and compute the cosine similarity between such vectors, obtaining a value between 0 and 1 (since term frequencies are always positive), where greater values represent greater similarity.

5. **Jaccard Similarity**. The Jaccard similarity [Jaccard, 1912] between two sets A and B is the ratio between the amount of common elements and of all elements, as shown in Equation 3.2.

$$\text{jaccard}(\text{A}, \text{B}) = \frac{|\text{A} \cap \text{B}|}{|\text{A} \cup \text{B}|} \tag{3.2}$$

   We consider the tokens in the text and the hypothesis as the input sets, and compute the Jaccard similarity on such sets to obtain a value between 0 and 1, where greater values represent greater similarity.

6. **Soft TF-IDF**. This feature combines Term Frequency–Inverse Document Frequency (TF-IDF) features and the Jaro-Winkler similarity metric [Winkler, 1990].

[1] `https://github.com/google-research/google-research/tree/master/rouge`

In the TF-IDF model, weights are assigned to words in a document (or sentence), such that the frequency of a word *t* in the document *d*, henceforth mentioned as *tf(t,d)*, is combined with the frequency of such word in the whole collection of documents. The latter frequency is named inverse document frequency, and is computed as the division of the total amount of documents by the amount of documents that contain word *t*, henceforth mentioned as *idf(t)*. TF-IDF is introduced in NLP textbooks [Jurafsky and Martin, 2009], and in general is defined as in Equation 3.3.

$$\text{tfidf(t, d)} = \text{tf(t, d)} \times \text{idf(t)} \tag{3.3}$$

To compute *idf(t)*, we consider a pair of sentences as the collection of documents, if this feature is computed for a single instance, or all the pairs of sentences in a target collection, if training a model.

We rely on the TF-IDF implementation provided by *scikit-learn*, where the *idf(t)* component is more complex than above described, and the TF-IDF values are normalized[2]. We employ the default configurations, where words are lowercased and accents are considered.

To compute this feature, each pair of words from the input sentences, with each word belonging to a different sentence, is input to the Jaro-Winkler metric[3]. If the resulting score is greater than the threshold of 0.9, than this feature accumulates the multiplication of such score and the TF-IDF of each word. Hence, the output of this feature is the result of a sum of values, one for each pair of similar words, according to the threshold on the Jaro-Winkler metric, and such that each value combines the Jaro-Winkler similarity score and the TF-IDF scores by multiplication.

Equivalence Features The features inspired on PI and RTE studies [Marques, 2015] are:

1. **NE Overlap**. The Jaccard similarity considering only words where either only the first letter is capital or all letters are capital.
2. **NEG Overlap**. The Jaccard similarity considering only language dependent negative words.
3. **MODAL Overlap**. The Jaccard similarity considering only language dependent modal words.
4. **BLEU**. This feature corresponds to the BLEU metric [Papineni et al., 2002b], which targets the overlap in n-grams of the input sentences, as computed with NLTK [Bird and Loper, 2004], and using the default of combining overlap scores up to 4-gram sequences.

   As BLEU combines the scores of the various n-gram overlap scores by multiplication, is possible that the BLEU score is 0 although overlap exists. For instance, short sentences may only have n-grams of lower order, hence the overlap score of the remaining n-gram orders would be 0, causing the BLEU score to be 0, even if overlap exists in any other n-gram order. To avoid such behaviour, and to avoid 0 values in this feature, we replaced its original configuration by employing BLEU with a smoothing function [Chen and Cherry, 2014].
5. **METEOR**. This feature corresponds to the METEOR metric [Denkowski and Lavie, 2014], which targets the overlap in unigrams, while considering word form variability.

   We replaced the original configuration of this feature to employ the latest version of the METEOR metric [Denkowski and Lavie, 2014], which includes more language specific options. Particularly, we configure METEOR for a target language when the input sentence representation is composed

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html`

[3] `https://pypi.org/project/jaro-winkler/`

of natural language, such as in a lowercased version of an original sentence. For instance, some languages support more word matching techniques than others[4], as for English and Portuguese respectively. For non supported languages, language independent techniques are employed, and we configure METEOR as such in the case of sentence representations where tokens are not actual words, such as when representing a sentence as a sequence of character trigrams.

6. **TER**. This feature corresponds to the Translation Edit Rate (TER) metric [Snover et al., 2006], which is similar to the Edit Distance feature, but also considers shifting words within a sentence.

7. **NCD**. This feature corresponds to the Normalized Compression Distance (NCD) metric [Li et al., 2004], which considers the input sentences generically as bytes, hence disregarding NLP. Its computation is based on the compressed (here relying on *bzip2*[5] compression) sizes of the input sentences, both isolated and concatenated.

8. **ROUGE-N**. The ROUGE-N metric targets n-gram overlap between a text and its hypothesis, as in BLEU, but considering the total number of n-grams in the text [Lin and Hovy, 2003].

   The original ROUGE-N computation was replaced by another framework[6].

9. **ROUGE-L**. A variation of the ROUGE metric based on the length of the LCS [Lin and Och, 2004].

   The original ROUGE-L computation was replaced by another, using the same framework employed for ROUGE-N.

10. **ROUGE-S**. A variation of the ROUGE metric based on skip-bigrams (i.e., bigrams of word tokens, allowing for in-between words) [Lin and Och, 2004].

Numeric Feature This feature addresses similarities between numbers in a sentence, by combining a similarity score on numbers only, with the similarity score of words that occur before and after numbers (a window of two words is considered). The similarity metric is Jaccard, and the combination is achieved by multiplication. Hence, the result of this feature is a real value between 0 and 1, corresponding to the multiplication of two Jaccard scores.

Text Representations The previously described features are applied to different representations of the sentences. We specifically considered the following representations:

1. **Original tokens**.

2. **Lowercased tokens**.

3. **Stems of lowercase tokens**.

4. **Word clusters**. The Brown clustering algorithm [Brown et al., 1992] is applied to a collection of language dependent documents, to group words into classes derived from the documents. Each word in a sentence is then replaced by its corresponding cluster identifier (a binary code), or 0 for words not in the vocabulary computed from the documents.

[4] `http://www.cs.cmu.edu/~alavie/METEOR/README.html`
[5] `https://www.sourceware.org/bzip2/`
[6] `https://github.com/google-research/google-research/tree/master/rouge`

Feature	O	L	S	C	DM	T
LCS	X	X	X	X	X	
Edit Distance	X	X	X	X	X	
Cosine Similarity	X	X	X	X	X	X
Length ratio	X	X	X	X	X	
Max Length	X	X	X	X	X	
Min Length	X	X	X	X	X	
Jaccard	X	X	X	X	X	X
Soft TF-IDF	X	X	X			
NE Overlap	X	X	X			
NEG Overlap	X	X	X			
Modal Overlap	X	X	X			
BLEU-3	X	X	X	X	X	X
METEOR	X	X	X	X	X	X
ROUGE N	X	X	X	X	X	X
ROUGE L	X	X	X	X	X	X
ROUGE S	X	X	X	X	X	X
TER	X	X	X	X	X	X
NCD	X	X	X	X	X	X
Numeric	X	X	X			

Table 3.1: Combination of features with representations, where O, L, S, C, DM and T correspond to Original, Lowercased, Stemmed, Cluster, Double Metaphone and Trigrams, respectively.

5. **Double Metaphone**. A well known algorithm to phonetically encode the words in the sentences, reducing words to a combination of 12 consonant sounds. The Double Metaphone (DM) algorithm [Philips, 1990] is based on English pronunciation, being more adequate to encode English words and foreign words often heard in the United States. However, we employ the same algorithm for all languages.

6. **Character trigrams**. Lowercased sentences are transformed into ordered sequences of tokens with 3 characters, corresponding to each character in the sentence and the following 2 characters.

Some features are not suitable to be combined with some representations, such as the numeric feature with the DM representation. The combinations can be seen in Table 3.1, where each feature corresponds to the application of the metric on the leftmost column to two sequences, built according to the lexical variants identified in the remaining columns. Such variants comprise lowercased (L) and stemmed (S) versions of the original (O) text. The cluster (C) and DM variants produce a sequence composed by non verbal codes, which:

- for cluster, correspond to binary strings that identify the cluster of each word, according to the Brown clustering algorithm [Brown et al., 1992] on a language dependent dataset,

- for DM, correspond to the codes of the DM algorithm for each word.

The trigrams (T) variant produces a sequence with a different length from the number of words in the original sentence, since it is composed by strings of 3 characters, one for each character in the original text.

3.2 Discourse Representation Structures

A formal semantic representation is a description of the meaning in a natural language sentence or segment, modelled in a logic based framework of symbols and predicates, and following a particular theory to analyse and model such meaning, regarding what are considered units of meaning and how these are arranged in a model. Such semantic representation is a source of features on semantic aspects, not available from lexical or syntactic analysis. In this work, we study the semantic representation of DRS, which implements the formal semantics definition of DRT [Kamp and Reyle, 1993].

For a pair of sentences, we compute a set of semantic features from the corresponding pair of DRS, which we designed to model semantic similarity, by leveraging equivalent aspects between the components on the DRS of each sentence, and such that the definition of equivalent aspects varies with the type of information in a component. The semantic features here defined are specific to DRS produced by the Boxer framework [Curran et al., 2007, Bos, 2008, Bos, 2015], which are only available for English. An example of a DRS generated by Boxer is shown in Figure 3.2, in a format of nested boxes that is suitable for human interpretation (Boxer also produces a logical form), and which we leverage to design our semantic features, as described in the following.

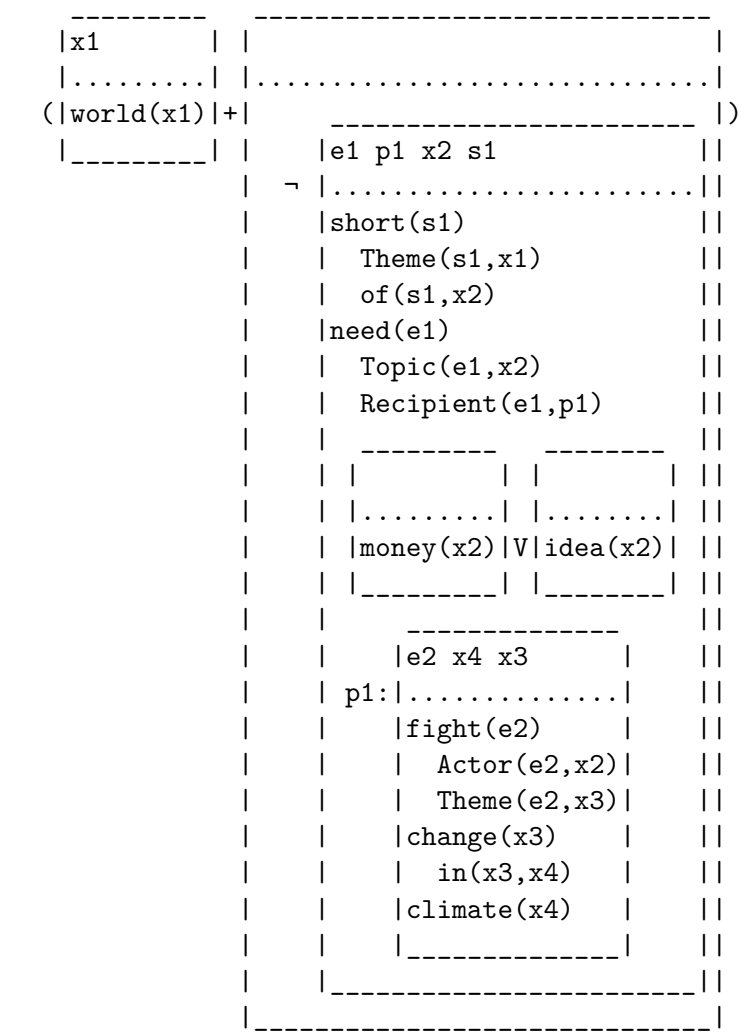


Figure 3.2: DRS for sentence "The world is not short of money or ideas needed to fight climate change.", as obtained from Boxer.

The formal semantics of DRT are modelled in DRS, essentially by a set of variables, called discourse referents, and a set of predicates, called conditions [Bos, 2008]. Such setup is further extended to represent more complex sentences, for instance by operators and groups of predicates, as shown in Figure 3.2, where symbols represent negation and the "or" conjunction, and multiple predicates are assigned to a single unit (variable "p1"). Following the DRT, a DRS is able to describe linguistic phenomena requiring meaning interpretation, such as anaphora and temporal expressions, and is suitable to model the meaning both in a single sentence and across sentences in a text, as implemented in Boxer [Bos, 2008].

Boxer is the last component in a pipeline of NLP components, that compose a framework to automatically produce semantic representations from a sentence or discourse segment [Bos, 2015]. An input is first split into tokens, following a particular form of representing certain tokens, such as abbreviations and parenthesis, for compatibility with the remaining components. The tokenized input is then parsed with the C&C tools [Curran et al., 2007], to produce a syntactic analysis in the form of a CCG [Clark and Curran, 2004], leveraging various resources such as POS and named entities. Boxer then composes a DRS from the CCG, as a logical form compatible with first-order logic [Bos, 2008].

Various works leveraged the first-order logic representation of a DRS from Boxer, for instance as a source of features to address semantic similarity tasks [Bjerva et al., 2014, van der Goot and van Noord, 2015b], or to employ DRS in other representation systems [Dakota and Kübler, 2016]. We instead extract features from the structure of predicates and symbols in the graphical representation of a DRS, since we designed our features from observation of DRS components, and not their translations as first-order logic formulas. Boxer is able to produce other semantic representations, such as AMR [Banarescu et al., 2013], but we target DRS only, which have fewer limitations on representing semantic phenomena, and hence are better suited to address more complex sentences [Bos, 2016].

To design our semantic features, we consider various views of a DRS, from the original form to simpler versions ignoring structure, and various types of information in predicates. For instance, some predicates contain words, either from the sentence or specific to the semantic analysis, which enable the computation of word similarity, and allow to define a semantic feature as the count of equivalent words between two DRS, eventually specific to a certain view of the DRS.

Some of our semantic features rely on components also available in DRS produced by other frameworks [Liu et al., 2018]. Moreover, our features rely on linguistic resources available in multilingual versions, such as word embeddings and WordNet [Bond et al., 2020], hence facilitating adaptation to frameworks producing multilingual DRS [Liu et al., 2021].

In Boxer, a DRS is provided in a logic format, based on the Prolog logic programming language, and in a graphical format that omits information from the logical form to facilitate visual interpretation. NLTK [Bird and Loper, 2004] implements an interface to Boxer, which provides an alternative graphical format, based on the logical form but containing more information than the graphical format from Boxer, and also additional annotations that unravel information encoded in the logical form. The semantic features here defined are specific to such NLTK representation.

3.2.1 DRS in NLTK

The NLTK graphical representation of a DRS, shown in Figure 3.5, is produced from the Boxer logical form shown in Figure 3.3, and contains more information from the logical form, such as POS information, than Boxer includes in its own graphical representation, shown in Figure 3.4.

Although the logical form contains the full semantic analysis, our features were designed from observation of the graphical representation, which is more complete and readable in the NLTK version. Also, some

```
sem(1,
[
   1001:[tok:'The',pos:'DT',lemma:the,namex:'O'],
   1002:[tok:woman,pos:'NN',lemma:woman,namex:'O'],
   1003:[tok:was,pos:'VBD',lemma:be,namex:'O'],
   1004:[tok:hospitalized,pos:'VBN',lemma:hospitalize,namex:'O'],
   1005:[tok:'June',pos:'NNP',lemma:'June',namex:'I-DAT'],
   1006:[tok:'15',pos:'CD',lemma:'15',namex:'I-DAT'],
   1007:[tok: (',pos: (',lemma: (',namex:'O'],
   1008:[tok:'Kansas',pos:'NNP',lemma:'Kansas',namex:'I-LOC'],
   1009:[tok:health,pos:'NN',lemma:health,namex:'O'],
   1010:[tok:officials,pos:'NNS',lemma:official,namex:'O'],
   1011:[tok:said,pos:'VBD',lemma:say,namex:'O'],
   1012:[tok:'.',pos:'.',lemma:'.',namex:'O']
],
merge(
drs([[1001]:x3],[[1002]:pred(x3,woman,n,0)]),
drs(
   [[]:p1,[]:e1,[]:x2,[]:x1],
   [
      []:prop(p1,drs(
         [[]:x4,[]:e2],
         [
            [1006]:timex(x4,date([]: (+),[]:'XXXX',[]:'XX',[1006]:'15')),
            []:rel(e2,x4,'Time',0),
            [1005]:timex(x4,date([]:(+),[]:'XXXX',[1005]:'06',[]:'XX')),
            []:rel(e2,x3,'Theme',0),
            [1004]:pred(e2,hospitalize,v,0)
         ])
      ),
      []:rel(e1,p1,'Topic',0),
      []:rel(e1,x1,'Actor',0),
      [1011]:pred(e1,say,v,0),
      [1010]:pred(x1,official,n,0),
      []:rel(x1,x2,of,0),
      [1009]:pred(x2,health,n,0),
      [1008]:named(x1,kansas,geo,nam)
   ]
))).
```

Figure 3.3: Boxer logical form for the DRS of sentence “The woman was hospitalized June 15, Kansas health officials said.”.

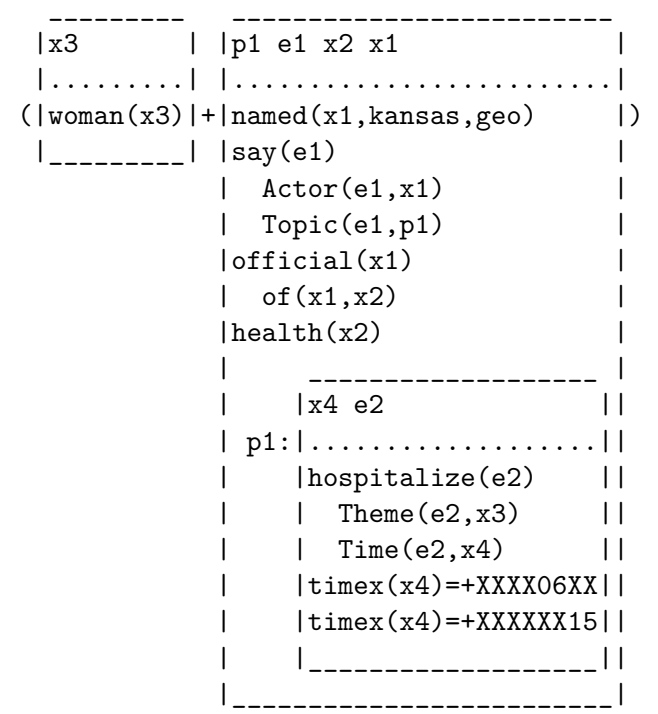


Figure 3.4: Boxer graphical representation for the DRS of sentence "The woman was hospitalized June 15, Kansas health officials said.".

information in the graphical representation from Boxer is encoded in a specific format, such as the "timex" predicates shown in Figure 3.4, that describe dates by using a sequence of characters. In NLTK, these predicates are represented in a logic format, as "a_date" predicates, parsed from the sequence of characters to explicitly mention the parts of a date, as shown in Figure 3.5.

In NLTK, a DRS is represented as a Python object, parsed from the corresponding logical form obtained from Boxer, and implemented with a visitor pattern, where the Python object is traversed by visiting each component in the graphical representation. The graphical representation from NLTK, shown in Figure 3.5, is produced from such Python object, and our features are implemented from a data structure, obtained by parsing each component in the graphical representation, to compute all features without traversing the Python object again.

The semantics represented in a DRS from Boxer, as observed in NLTK, may not correspond to the human interpretation of the source sentence, both due to failure of the Boxer analysis, or due to the transformations applied by NLTK. Some of these misinterpretations are related to sentence parsing, and may affect the structure of a DRS, for instance when predicate groups are created for clauses of a sentence that were erroneously detected. As such, some of our features consider a DRS as a single set of predicates, thus ignoring the structure introduced by predicate groups. For instance, the predicates within the inner partition illustrated in Figure 3.5, identified by variable "p1", are considered separately in some features, while other features ignore the partition and consider such predicates as any other.

Given a pair of sentences, the corresponding pair of DRS is obtained from Boxer using the NLTK interface, and a set of features is essentially composed from: a) identifying semantic aspects in any DRS (boolean features), b) counting aspects of a DRS that have an equivalent in the other (count based features), or c) calculating distances between numbers occurring in each DRS. The components required to compute such features are described in the following, as available in the NLTK version of a DRS from Boxer. To

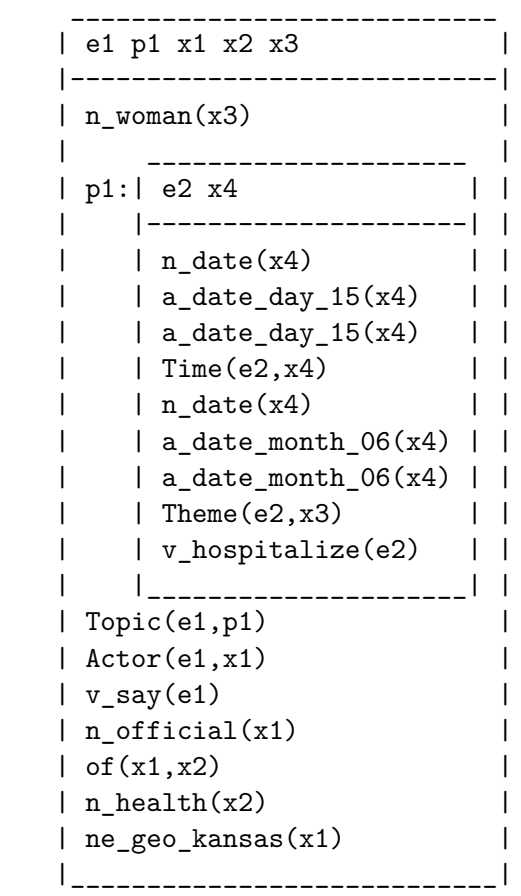


Figure 3.5: NLTK graphical representation for the DRS of sentence "The woman was hospitalized June 15, Kansas health officials said.".

compute the DRS, we configured Boxer to transform expressions with equality symbols into a logical form (eliminate equality), and resolve referential expressions such as pronouns. Although Boxer provides other configuration options [Bos, 2015], not all are available in the NLTK interface.

3.2.2 DRS Components

Our features are drawn from the representation of a DRS as a set of referents and conditions recursively organized into boxes, and some of our features leverage the depth of such boxes. For instance, the DRS shown in Figure 3.5 contains an inner box, which follows the same format as the box of the main DRS. This inner box represents a different depth of analysis, and is a (inner) DRS for a particular segment of the sentence (namely, "was hospitalized June 15"), being labelled with a single referent ("p1") to mention such inner DRS in conditions of the main DRS. Some of our semantic features leverage depth information, for instance by computing the count of equivalent words from conditions in boxes of the same depth.

Referents correspond to entities, explicit from the sentence, or implicit as required to represent the semantics of the sentence. For instance, in the DRS shown in Figure 3.5, the referent "x3" corresponds to word "woman", while referent "p1" corresponds to an inner DRS for a clause of the sentence. A referent is an identifier, randomly generated and unique within the main DRS of a given input, that declares an entity within the scope of a DRS. Conditions declare, detail or relate referents, and are represented as a predicate, composed by a functor and N arguments.

Referents are assigned to entities using a unary condition, named a property [Bos, 2008], where the functor encodes textual information about the entity. If the entity is a word in the sentence, the functor begins with its POS, ends with its lemma, and in between it may include additional details. For instance, in Figure 3.5, the word "Kansas" is represented by the condition "ne_geo_kansas(x1)", where the argument "x1" is declared as its referent, and the functor "ne_geo_kansas" details "Kansas" as a named entity ("ne")) of geographic type ("geo"). Entities may have multiple properties, which are represented by using the same referent on multiple unary conditions, as seen in Figure 3.5 for "Kansas (health) officials", which is considered a single entity with properties "ne_geo_kansas(x1)" and "n_official(x1)".

Inner DRS are a mean to manipulate groups of predicates as frames with self contained semantics, such as to represent negation or implication. A negated clause in a sentence is represented by an inner DRS with predicates expressing the clause without negation, and a symbol attached to the box to indicate the negation of its contents. For instance, in the example of Figure 3.6, "not (been) hospitalized" is represented by an inner DRS with a property for the verb "hospitalize", and a symbol next to the DRS box to indicate that the action implied by such verb does not hold.

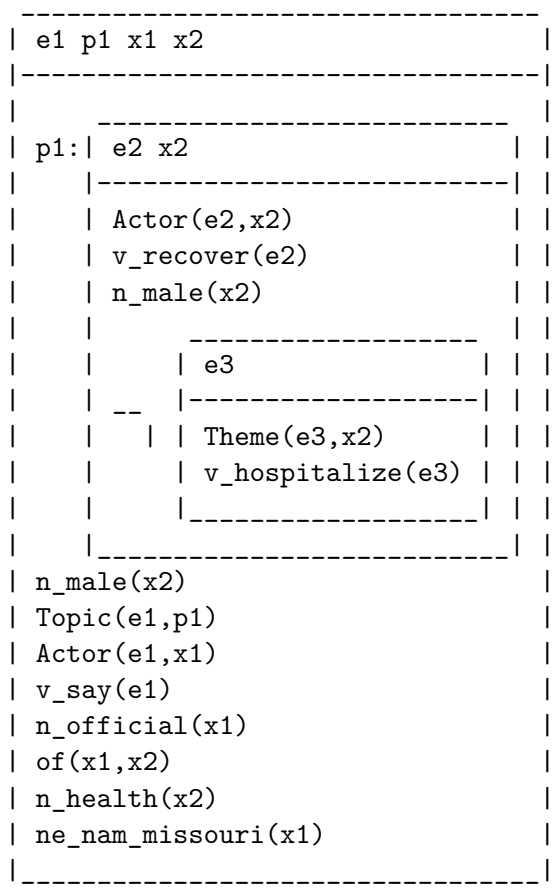


Figure 3.6: The DRS for sentence "Missouri health officials said he had not been hospitalized and is recovering.", as obtained from the NLTK interface to Boxer. This sentence is part of a non paraphrase example from the MSRP corpus, where it is paired with the sentence mentioned in Figure 3.5.

In the example of Figure 3.6 is shown a repetition of predicates in different DRS depths, where property "male(x2)", which corresponds to the pronoun "he" in the sentence, exists in the main DRS and also in the inner DRS. If we replace "he" by "the man" (or a name), the duplicated property is replaced by a single property in the main DRS, and the referent of such property is still properly mentioned in the inner

DRS, as shown in the example of Figure 3.7.

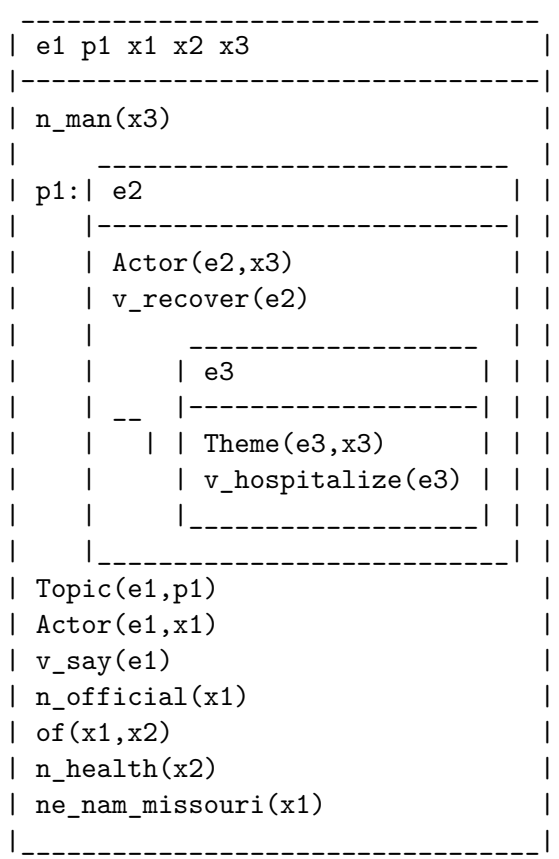


Figure 3.7: The DRS for sentence "Missouri health officials said the man had not been hospitalized and is recovering.", which is based on the example shown in Figure 3.6, but replacing "he" by "the man".

For implication, two inner DRS are connected by a symbol indicating direction ("->", in NLTK), to represent that one inner DRS is conditioned by the other. For instance, implication would arise from employing "then" in a sentence, such as to link an action to its consequence.

In conditions with multiple arguments, the functor is a token specific to the DRS, indicating the semantic relation between the argument referents. For instance, in Figure 3.5, "health officials" is represented by condition "of(x1,x2)", where "x1" has a property with functor "n_official" and "x2" has property "n_health", to ultimately represent "health officials", which could be interpreted as *official of health*. Binary conditions may also describe semantic roles, as shown in the condition "Actor(e1,x1)" illustrated in Figure 3.5, which indicates that the entity of referent "x1" triggers the action assigned to referent "e1", that corresponds to a form of the verb "say".

Numbers are parsed, normalized to a fixed representation, and preceded by their type. For instance, "$ 693 million" is represented by functor "card_eq_693000000". For dates and times, specific fragments are distinguished, and arranged into different properties. For instance, in Figure 3.5, "June 15" is represented by two properties, with functors "a_date_day_15" and "a_date_month_06", both assigned to referent "x4" to form a single entity from such multiword expression.

Using the previously mentioned components, in the following we describe: a) boolean features, such as to indicate the presence of a negation in any DRS, b) count based features, as to represent the number of entities in a DRS that have an equivalent in the other, c) percentage based features, as for the ratio of

equivalent entities in all possible entity pairings, and d) distance based features, such as from measuring the mean gap between dates in one DRS to dates in the other.

3.2.3 Features From a Pair of DRS

One of the forms we employ to produce features from a pair of DRS is to select a set of unary conditions from each, according to a certain form of parsing conditions, and count how many pairs of such conditions are related, according to a certain form of comparison. Hence, from a selection of pairs of DRS properties and a form of assessing their relatedness, we compute a pair of features, namely a count of how many such conditions are related, as a discrete value greater than 0, and a percentage from such count, as a continuous value between 0 and 1. Moreover, if two unary conditions are related as equivalent, then their referents are also considered equivalent, and we leverage such information to produce features. In the following, we describe the computations that support our features, such as how to organize conditions, forms of comparing conditions, and how to leverage other DRS components, such as negations and binary conditions.

Conditions by Referent and Depth Relatedness of unary conditions is assessed from their functors, and a feature is computed as the count of related functors between two DRS. To leverage the referent and depth in unary conditions, we define various forms of organizing the conditions of a DRS, in collections indexed by referent and/or depth, where to each index corresponds a set of functors, as shown in Figure 3.8.

Features addressing relatedness of unary conditions are computed between indexes (or subindexes), from collections of the same type. The result is a sum of partial results, from computing the feature on the various indexes of a collection. Namely, the functors on an index of a DRS are compared with the functors on an index of the other DRS, and each index of a DRS is compared with all indexes of the other DRS. The value of a feature is then the total count of pairs of unary conditions considered related.

Feature values for different collections are similar, and often identical, but allow to consider different views of a DRS. For instance, the duplicate unary condition , previously shown in Figure 3.6, is seen as a single condition in the collection organized by referent, but is preserved as duplicate in the collection organized by depth. The different types of collection differ according to their indexing method, and are described in the following, also with a description of how the duplicate property is represented in each collection. Namely, collections are indexed by:

- referents, where to each referent identifier corresponds a set of functors. The duplicate property example, from Figure 3.6, is here represented by a single entry, of functor “n_male” on index “x2”. This collection allows to distinguish identical functors assigned to different referents.

- depths, where to each inner DRS corresponds a set of functors. The depth of the main DRS is 0. The functor of the duplicate property, in the previously mentioned example, is here represented twice, in the set for depth 0 and again in the set for depth 1.

- both of the above, such that to each depth corresponds a set of referents, and to each of such referents corresponds a set of functors. The duplicate property is here represented twice, as in the collection based on depths, but preserving the referent identifier, as in the collection based on referents. This collection preserves more information and structure of the DRS than any other.

Collection by referent

```
--index x2
n_health
n_male

--index x1
n_official
ne_nam_missouri

--index e1
v_say

--index e3
v_hospitalize

--index e2
v_recover
```

Collection by depth

```
--index 0
n_male
v_say
n_official
ne_nam_missouri
n_health

--index 1
v_recover
v_hospitalize
n_male
```

Collection by both referent and depth

```
--index 0

---subindex x2
n_male
n_health

---subindex x1
n_official
ne_nam_missouri

---subindex e1
v_say

--index 1

---subindex x2
n_male

---subindex e3
v_hospitalize

---subindex e2
v_recover
```

Figure 3.8: Different forms of organizing unary conditions, by referent and/or depth, for the DRS shown in Figure 3.6. The collection organized by neither referent nor depth (not shown) is a set with all functors in any of the remaining collections, without duplicates.

- none of the above, which corresponds to a set of all functors, from all referents and depths. The duplicate property is here represented by a single entry for functor "n_male". Repeated functors with different referents are also represented in a single entry, since the set does not contain duplicates.

Relatedness Tests As previously mentioned, some features are computed from counting how many unary conditions are related between two DRS, as assessed from their functors, according to various forms of computing relatedness. Namely, a pair is formed with a functor from each DRS, named A and B in the following, and a counter is incremented when these are related as either:

- antonyms, if any of the antonyms of A is B, or a synonym of B, according to antonyms and synonyms from the WordNet [Fellbaum, 1998] database.
- similar, if A is similar to B, according to a sequence of tests leveraging various resources, as below described.
- reachable in WordNet, if the average length of hypernym paths between A and B, and any of their synonyms, in the WordNet network, is less than 10 hypernym nodes.

- functor match, if A is identical to B. This test simultaneously compares lemmas and annotations/symbols from the DRS analysis, such as POS and type of named entity.

Except for the functor match test, functors are transformed into natural language words, to enable a match in lexical resources. For instance, "ne_geo_kansas" becomes "kansas".

As an example of a feature addressing relatedness of unary conditions, using the functor match test on the collections indexed by depth shown in Figure 3.9, we obtain a feature with value 4, corresponding to the match of functors "v_say", "n_official" and "n_health" on depth 0, and "v_hospitalize" on depth 1.

DRS A

```
--index 0
n_woman
v_say
n_official
n_health
ne_geo_kansas

--index 1
n_date
a_date_day_15
a_date_month_06
v_hospitalize
```

DRS B

```
--index 0
n_male
v_say
n_official
ne_nam_missouri
n_health

--index 1
v_recover
v_hospitalize
n_male
```

Figure 3.9: Unary conditions indexed by depth, for the DRS of sentence "The woman was hospitalized June 15, Kansas health officials said ." (DRS A) and of sentence "Missouri health officials said he had not been hospitalized and is recovering ."(DRS B), previously shown in Figures 3.5 and 3.6 respectively.

The similarity relatedness test is implemented as an ordered sequence of tests involving various similarity metrics and resources for word similarity. Namely, and in the following order of testing, two words are:

- similar, if the words exist in the vocabulary of counter-fitted embeddings [Mrkšić et al., 2016], and the cosine of such embeddings is greater than 0.5. This threshold was chosen by observation. Counter fitted embeddings enhance word similarity from corpora based word embeddings by leveraging synonym and antonym information, at the cost of a smaller vocabulary than purely corpora based embeddings.
- not similar, if any of the antonyms of a word is a synonym of the other, according to antonyms and synonyms from the WordNet [Fellbaum, 1998] database.
- similar, if any of the synonyms of a word is in the synonyms of the other, according to WordNet.
- similar, if there is a pair of WordNet synonyms, one for each word, that exists in the vocabulary of counter-fitted embeddings [Mrkšić et al., 2016], and the cosine of such embeddings is greater than 0.5.
- similar, if there is a pair of WordNet synonyms, one for each word, for which the cosine of their FastText embeddings [Bojanowski et al., 2017] is greater than 0.6. This threshold was chosen by observation, to consider words that is possible to envisage as similar, while preventing words that are not plausible to be considered similar in natural language. FastText is not limited by a vocabulary,

as it is based on characters, but we lower case all words and do not consider words containing punctuation symbols, as we observed that the cosine of embeddings for words with these factors is different from that of the corresponding normalized words.

For each of the relatedness tests, we compute ten features, where five correspond to counts and the remaining correspond to percentages from such counts. Of the five count based features, four are based on applying a relatedness test to each type of collection. The remaining count based feature corresponds to the number of pairs of referents from unary conditions considered related, according to the relatedness test. Since the same referent may correspond to various unary conditions, this latter count may differ from counts based on functor relatedness. For instance, if a pair of referents, one from each DRS, corresponds to two unary conditions in each DRS, both considered related, the count of related referents is 1, while the count of related functors is 2.

The percentages for counts based on relatedness tests are computed by dividing a count of functor pairs considered related by one of the relatedness tests, by the total number of functor pairs tested. The percentage for the count based on related referents is computed by dividing the count of referent pairs considered equivalent by the total number of possible referents pairs, obtained by multiplying the total number of referents in one DRS with the total number of referents in the other.

Referents of unary conditions with related functors are considered equivalent, except for those resulting from the antonym relatedness test. The equivalent referents from the various combinations of relatedness tests and collection types are aggregated into a common set.

Table 3.2 summarizes the outputs computed from unary conditions and relatedness tests, where columns are relatedness tests performed, and rows are types of output on such tests. For instance, *referents (count + percentage)* indicates that two features (count and percentage) are computed from the collection of functors based on referents, and *equivalent referents (set)* indicates which related referents are considered equivalent. Each mark indicates that the row element is computed when testing for the correspondent relatedness. All marks represent two features, mentioned in the row (count and percentage), except for those of the last row, that represent the contents of the set for equivalent referents. As such, 40 of our features are displayed.

	antonyms	similarity	WordNet distance	functor match
referents (count + percentage)	X	X	X	X
depths (count + percentage)	X	X	X	X
both (count + percentage)	X	X	X	X
none (count + percentage)	X	X	X	X
related referents (count + percentage)	X	X	X	X
equivalent referents (set)		X	X	X

Table 3.2: Outputs from unary conditions and relatedness tests.

Binary Conditions Using the set of equivalent referents computed from unary conditions, we also compute a set of equivalent referents from binary conditions, to produce two features, for the count and percentage of such equivalent referents. To design these features, we focused on binary conditions that describe semantic roles, where the left referent corresponds to an event (a verb in the sentence), and the right referent corresponds to an entity that triggers the event. However, our method is applied to all binary conditions.

We search for a pair of binary conditions, one from each DRS, where the pair of left referents is in the

equivalent referents previously identified from unary conditions. Upon finding such match, if both binary conditions have identical functors (to ensure we compare the same type of relation), we consider the pair of right referents as equivalent. For instance, in the pair of DRS shown in Figures 3.5 and 3.6, both contain a binary condition with functor "Theme", where the left referents ("e2" and "e3") form a pair that is part of our set of equivalent referents from unary conditions, as both correspond to functor "v_hospitalize", which represents the verb/action "hospitalize". Hence, we consider the right referents "x2" and "x3" equivalent, which correspond to functors "n_male" and "n_woman" respectively. Such equivalency is plausible for our example, since "Theme" is a semantic role to denote an entity affected by an action/event, as introduced in NLP textbooks [Jurafsky and Martin, 2009], and both sentences target the "hospitalize" event, one relative to a woman and the other relative to a man.

Some of the pairs of equivalent referents obtained from binary conditions may already be part of the set of equivalent referents from unary conditions, but we include them in the set computed from binary conditions, to represent that the referents are considered equivalent by different methods. From the set of equivalent referents from binary conditions result two features, a count of its items and a percentage of such count, computed as in the related referents from unary conditions. The equivalent referents from binary conditions are joined with the equivalent referents from unary conditions to form a full set of equivalent referents.

Properties Following the procedure previously employed for unary conditions, we also compute features regarding entities with multiple properties, to model the amount of similarity between entities/referents declared in more than one condition, hence more detailed in the DRS. Namely, using the collections indexed by referents only and by both depths and referents, we compute counts (and percentages) of functors considered related between referent indexes containing more than one functor, using all relatedness tests except for antonyms.

The main difference to the previous features from unary conditions is that we now only consider referent indexes containing more than one functor. For instance, using the functor match test on the collections indexed by referent, shown in Figure 3.10, we obtain a value of 1 for the correspondent count based feature, since for any pairing of groups (each from a different DRS) with more than one functor, the only functor that exists in both groups is "n_official". To compute the percentage based feature, the count is divided by the total number of functor pairs from indexes with more than one functor.

Using the full set of equivalent referents and the collection indexed by referents only, we also compute counts and percentages, to represent the amount of similarity between entities with multiple properties that correspond to referents previously considered equivalent. Namely, for each pair of equivalent referents, we obtain the functors of each referent, and if both have more than one functor, we compute counts and percentages for all relatedness tests except for antonyms. Table 3.3 summarizes the features computed from entities with multiple properties.

	similarity	WordNet distance	functor match
referents (count + percentage)	X	X	X
both (count + percentage)	X	X	X
equivalent (count + percentage)	X	X	X

Table 3.3: Features from entities with multiple properties. Columns are relatedness tests performed, rows are collections involved on such tests. For instance, *referents (count + percentage)* indicates that two features (count and percentage) are computed from the collection indexed by referents only. Each mark indicates that the row element is computed when testing for the correspondent relatedness test. All marks represent two features, mentioned in the row, namely count and percentage. As such, 18 of our features are here displayed.

DRS A

```
--index x3
n_woman

--index x1
n_official
ne_geo_kansas

--index x2
n_health

--index x4
n_date
a_date_day_15
a_date_month_06

--index e1
v_say

--index e2
v_hospitalize
```

DRS B

```
--index x2
n_health
n_male

--index x1
n_official
ne_nam_missouri

--index e1
v_say

--index e3
v_hospitalize

--index e2
v_recover
```

Figure 3.10: Unary conditions indexed by referent, for the DRS of sentence "The woman was hospitalized June 15 , Kansas health officials said ." (DRS A) and of sentence "Missouri health officials said he had not been hospitalized and is recovering ."(DRS B), previously shown in Figures 3.5 and 3.6 respectively.

Negation and Implication Our features also target the negations and implications in DRS, by comparing these components between both DRS of an example. For instance, if both DRS contain a negation component, we compute a feature from the count of equivalent conditions between the respective inner DRS, and another feature from the percentage of such count. No distinction is made between multiple negations or implications in a DRS, hence conditions from multiple negations/implications are considered as a single set for each DRS.

To compute equivalency, we leverage the previously computed set of equivalent referents. Namely, we consider two unary conditions from either negations or implications, one from each DRS, as equivalent if the corresponding pair of referents is in the set of equivalent referents. A pair of binary conditions is considered equivalent if their functors are identical, and both the pair of referents composed by left referents only and the pair with right referents only are on the set of equivalent referents.

We produce one feature with the number of matching conditions (unary and binary) between negations, and another such feature from implications. The percentages of such counts are also features, where the count is divided by the total number of conditions in negations/implications, as obtained by multiplying the total number of conditions in the negations/implications of one DRS with the total number of conditions in the negations/implications of the other. As such, we produce four features, two counts and two percentages, for the equivalency between unary and binary conditions in negations/implications.

Other than count based features, we also produce 4 binary features, corresponding to the presence of negation or implication in each DRS of an example pair. For instance, one of these features models if the left DRS has a negation component, and is set to 1 if true and 0 otherwise. As such, we allocate four

features to model the existence of a negation/implication in the left/right DRS.

Numbers and Dates As previously mentioned, semantics for numbers are specified in DRS, for instance by using specific functors to distinguish parts of a date, as shown in Figure 3.5. To leverage such semantic information, we designed features based on the similarity between numbers in each DRS of an example pair, considering conditions relative to dates separately from conditions for other numbers. Namely, we compute a pair of features regarding dates and a pair of features regarding other numbers, where one of the features in either pair considers a flat view of DRS, with all conditions in a single set, and the other considers the original DRS form, with conditions eventually arranged in various sets to represent inner DRS from various depths.

To compute a single value for all numbers of a certain type in a set of conditions, we first parse numbers from the text of functors. The numbers are then aggregated into two sums, corresponding to the two types of numbers we consider in our features, as previously mentioned. Namely, one of the sums represents all the numbers from dates, as parsed from functors that begin with "a_date", and the other represents the remaining numbers, parsed from functors that begin with "card_eq". Using such sums, we compute the mean value for numbers of a certain type, by dividing a sum by the corresponding amount of considered numbers. For instance, all numbers from parts of a date, and from multiple dates, are summed to produce the mean value regarding dates.

For features leveraging the flat view of DRS, the feature value for a certain type of number is computed as the absolute difference between two means, one from each DRS. For features leveraging the original DRS form, a mean is computed for each depth, and a global mean of means is computed to aggregate the numeric information from all depths, such that to each DRS corresponds a single mean value, as in the flat view. Then, the feature value for a certain type of number is also computed as the absolute difference between two means, one from each DRS.

3.3 Embeddings

We obtained embeddings from BERT [Devlin et al., 2019], namely a single embedding for the concatenation of two target sentences, by employing BERT models not tailored to a particular task, since in early experiments these achieved greater performance than fine tuned BERT models. From such a generic embedding of a sentence pair, as we will see, we can build a model through supervised learning with non-deep-learning methods.

The BERT architecture defines two variants in model size, with 12 layers and embeddings of 768 dimensions for BERT-Base models and 24 layers with embeddings of 1024 dimensions for BERT-Large models. The output of a BERT layer is a sequence of embeddings for input words and special tokens of the BERT architecture, and each layer encodes different linguistic information [Jawahar et al., 2019, Tenney et al., 2019, Clark et al., 2019]. As such, the embedding of an input text is obtained from the output of a certain layer, either from its *CLS* special token, intended for classification purposes [Devlin et al., 2019], or by reducing the sequence into a single embedding, such as by a weighted average.

A generic BERT model is obtained by pre-training, a process where the weights of layers are adjusted according to two unsupervised language modeling tasks, applied to a collection of unlabeled texts [Devlin et al., 2019]. One such unsupervised task randomly masks some of the input tokens, and then finds the original forms of the masked tokens, according to a predefined vocabulary. The final BERT model is then defined by a network of pre-trained weights and the vocabulary, and its output is a generic representation of the semantics in the input text, according to pre-training data.

Given a pair of sentences, the essential input to a BERT model is a sequence of indexes on its vocabulary, one for each token in the concatenated sentence pair. For sentence pairs, BERT defines an optional second input, as a mask vector that identifies which tokens belong to which sentence. Moreover, BERT defines a final optional input as a mask vector that identifies padding tokens to discard, which we employed to limit our input sequences to a fixed length of 128 tokens, by padding or truncating sequences of other lengths, following the original BERT configuration (`https://github.com/google-research/bert`). All of our models consider raw text that was not transformed by lower casing or accent removal.

We operated generic BERT models with the bert-as-service framework (`https://github.com/hanxiao/bert-as-service`), where the full BERT input is automatically computed from a sentence pair concatenated by the "|||" separator. The default output is computed from the second to last layer, using a weighted average of token embeddings, normalized by the padding mask to consider only non-padding tokens. This layer has shown to contain better embeddings [Liu et al., 2019b].

3.4 Models

Given a corpus where each example is composed by a pair of sentences, and its target outcome is a label and/or continuous value related to the equivalence between the two sentences, our (traditional) models are based on two distinct types of feature vector: (a) the embedding of each sentence pair according to BERT, and (b) a vector of scores from similarity metrics. Using BERT embeddings, we also build models based on fine tuning the embeddings (BERT fine tuned) to tasks in a particular corpus, in an end-to-end fashion, where BERT is tailored to the tasks in a corpus.

3.4.1 Traditional Models

In this thesis we employ corpora annotated with PI labels, NLI labels or STS scores. Therefore, we compute classification and regression models for each type of feature vector, such that the same type of features is evaluated in all tasks. We also combine different types of feature vector into a single vector, which is also evaluated in all tasks. Namely, we compute seven models for each experiment, corresponding to all combinations between lexical features, DRS features and BERT embeddings. These models are built with traditional machine learning algorithms, such as SVM, rather than neural networks. All machine learning was performed in *scikit-learn* [Pedregosa et al., 2011].

Classification models were chosen in various combinations, mainly by considering learning algorithms available in what we call *simple* and *complex* versions, and the processing times to build them. For instance, we consider that the simple version of SVM has a linear kernel (LIBLINEAR implementation), while the complex versions correspond to the non-linear polynomial and Radial Basis Function (RBF) kernels. Also, a random forest is a combination of decision trees and random feature selection, hence a complex version of decision trees. Hence, the final set is composed of three types of SVM, corresponding to different kernels, decision trees and random forests of decision trees. In addition, we considered an ensemble of all models based on a voting algorithm.

To enforce reliable prediction probabilities, all classifiers were calibrated [Zadrozny and Elkan, 2002], using the Platt method [Platt, 2000], as implemented in *scikit-learn*.

Regression models follow the same types of algorithms selected for classification models, and were also combined with a voting model, as further described in the following.

Voting Model For both classification and regression, we employed a voting strategy on a set of different models, to leverage different learning strategies at once. For regression, voting consists of averaging predictions from a set of models, while for classification different strategies of computing the output class may apply, whether by choosing the class predicted by most classifiers or by averaging the prediction probabilities that each classifier reports for a certain class, and choosing the class with the highest average.

We considered that all models have the same weight, and for classification models we employed a strategy (named "soft voting" in *scikit-learn*) in which the output class was chosen by averaging the prediction probabilities that each classifier reported for a certain class, and choosing the highest class average using argmax.

Scaling and Normalization As assessed from our experiments, machine learning algorithms perform better when features are transformed to a common scale, while retaining their magnitude and data properties. As such, we define a set of scaling and normalization operations for certain types of features, to ensure that all feature values are between -1 and 1. The only processed features are embeddings and count based features, since the remaining features already produce values within such scale. Scaling and normalization operations are employed on all feature vectors, both of a single type of features and of feature combinations, and are computed with tools provided by *scikit-learn* [Pedregosa et al., 2011], as described in the following.

Originally, the features that compose BERT embeddings vary approximately between -25 and 2, with a mean value close to 0, as computed with maximum, minimum and mean functions on a set of embeddings, for some of our target corpora. We convert features in BERT embeddings to a -1 to 1 scale with the *MaxAbsScaler* from *scikit-learn*, which is most suitable for sparse data, where each feature is converted individually such that the maximum absolute value in a set of embeddings (for instance, the train or test splits) is 1, hence retaining data properties, such as sign and sparsity.

Count based features are defined to have only positive values, but their maximum range is unlimited. For instance, in the test set of the SICK corpus, the maximum feature value is 30 for lexical features and 207 for DRS features. For such features, we compose a pipeline of transformations, using default configurations, such that all discrete feature values from counts are transformed into continuous values between 0 and 1. We consider features with values greater than 1 as count based features, and select the sequence of transformations based on early experiments. First, we transform count based features with *RobustScaler*, which is a tool focused on addressing outliers, that most transforms feature values, by centering data according to the median value, and scaling according to quantiles. The next transformation is employed by *PowerTransformer*, which approximates data to a normal distribution. Lastly, we employ *MinMaxScaler*, to enforce that feature values are between 0 and 1.

3.4.2 BERT Fine Tuned

We define BERT fine tuning as a multi-input and multi-output neural network that encompasses the BERT architecture. In our fine tuning architecture, BERT is followed by either one neural output layer, for PI, or two neural output layers, one for NLI classification and the other for STS regression, for corpora where each example is described relative to both of these tasks. As such, the inputs of our fine tuning model are the same as those of the BERT model, namely, the indexes and masks computed from a sentence pair, as previously described. The outputs are a similarity value, for the STS task, and a probability distribution for the target classes of the NLI or PI tasks.

Following the original BERT fine tuning setup [Devlin et al., 2019], the weights of our task specific layers were initialized from a truncated normal distribution, and their input was the embedding of the

CLS special token in the last layer of the BERT model, further normalized by a non-linear tanh-based layer (https://github.com/google-research/bert/issues/43) and processed by a dropout layer. For corpora that describe both a classification and a regression task, the activations of our task-specific layers are accordingly softmax and linear, where softmax corresponds to reducing the features in the BERT embedding into a probability distribution for the classes of the classification task in a target corpus, and linear corresponds to reducing the BERT features into a single value.

In fine tuning, the layers concerning the BERT model are initialized with pre-trained weights, which are adjusted according to the loss between the outputs of the classification and regression layers and the correspondent labels in a target corpus. Namely, for regression we employed the mean absolute error loss, since it provides robustness to outliers, as convenient in applying our model to diverse corpora. For classification, we employed the categorical cross entropy loss, which allowed us to address binary and multi-class tasks with the same setup.

Due to the dropout layer, which randomly ignores a different part of the BERT output on each training of the same model, our network outputs are non deterministic. Hence, all reported results for fine tuned models correspond to an average of five instances of each model.

The fine tuning architecture was defined with the Keras framework (https://keras.io/). Loading the BERT model and preparing its inputs from a sentence pair was performed with the Transformers toolkit (https://huggingface.co/transformers/).

3.5 Summary

In this chapter we described the underlying features of systems developed in this thesis, and the modelling techniques employed with such features.

In the next two chapters we present experiments for the English and Portuguese language, respectively, where both describe the experimental setup, such as for evaluation metrics and corpora, and discuss the particularities of obtained results.

4

Evaluation on the Impact of Semantic Features for English

To assess the similarity between English sentences, we designed models that leverage BERT embeddings for English only, lexical similarity metrics, and semantic features from DRS, separately and combined. The experimental setup to evaluate our models is described in the following sections, where we also discuss the obtained results. Namely, we evaluate models supervised on corpora suitable for the tasks of PI, NLI and STS, employing the corresponding test sets. Our models include fine tuning BERT embeddings on such corpora, using generic BERT embeddings in traditional models, for instance based on SVM, and also combining such embeddings with lexical and/or DRS features. For this evaluation, we consider both DRS features and BERT embeddings as semantic features.

4.1 Experimental Setup

In this section, we describe the computation of language dependent features, the configuration of our models, the corpora where the performance of our models is assessed and the evaluation metrics employed.

4.1.1 Lexical Similarity Features

The language dependent lexical features were adapted to English as described in the following.

To compute word clusters, previously described in Section 3.1, we employed the Brown clusters of the Yelp Academic Dataset, as provided in [Schneider et al., 2014].

Negative words, required by the NEG Overlap feature previously described in Section 3.1, are defined as *not*, *no*, *never*, *nothing*, *none*, *nobody*, following [Marques, 2015].

Modal words, required by the MODAL Overlap feature previously described in Section 3.1, are defined as *can*, *could*, *may*, *might*, *will*, *would*, *must*, *shall*, *should*, *possible*, *possibly*, following [Marques, 2015].

4.1.2 BERT Embeddings

We employed the base and large versions of the English BERT models (BERT-Base and BERT-Large) described in `https://github.com/google-research/bert#pre-trained-models`. In all BERT models, we employ raw text that was not transformed by lower casing or accent removal.

For experiments involving traditional models, we obtained a single embedding for the concatenation of the two target sentences in an example, from generic BERT models not tailored to a particular task, such that the embedding is considered as a set of features for the example.

4.1.3 Model Configuration

We followed the fine tuning parameter recommendations from [Devlin et al., 2019], such as for the range of values for epochs, batch size and learning rate, but did not perform automatic search for optimal parameters. Instead, we selected the maximum number of recommended epochs and batch size, respectively 4 and 32, and the intermediate value for learning rate (3×10^{-5}). For the optimizer, we employed Adam [Kingma and Ba, 2015], since in early experiments this was the best setting, unlike the original BERT model which employs a version of Adam featuring weight decay [Devlin et al., 2019].

For all traditional models, optimal parameters were identified from a combination of various parameters, including various degrees for the polynomial kernel, the number of decision trees in random forests and the existence of class imbalance on all classification models. Said parameter search was applied for each corpus. For instance, to obtain the final model, for a certain corpus and feature set, when using SVM with a linear kernel, seven different models were trained, corresponding to different values for the C parameter, sampled from a logarithmic scale between 0.001 and 1000. When using SVM with the remaining kernels, the search included at least the C and gamma parameters, such that each of the mentioned seven models implies training another set of models, corresponding to combinations of a certain C values and various values for gamma, which were sampled from a logarithmic scale between 0.0001 and 10. For random forests, various types of parameters were also included in the search, such as the number of trees (we experiment with 100 or 200 trees) and the maximum depth of each tree.

4.1.4 Corpora

Microsoft Research Paraphrase Corpus

For the evaluation of the PI task we rely on the MSRP corpus [Dolan and Brockett, 2005], previously introduced in Section 2.3. In this corpus, each example is composed by two sentences and a numeric value, which is 1 if the sentences are a paraphrase and 0 otherwise. Examples of paraphrases are shown in Figure 4.1, while non paraphrases are shown in Figure 4.2, where the first and second sentences are identified respectively as 1 and 2 to represent the order presented in the corpus, although order is irrelevant in PI.

```
Sentence 1: More than half of the songs were purchased as albums, Apple said.
Sentence 2: Apple noted that half the songs were purchased as part of albums.

Sentence 1: Powell fired back: "He's accusing the president of a ludicrous act,"
he said.
Sentence 2: If so, Powell said, he's calling the president ludicrous, too.
```

Figure 4.1: Examples of paraphrases from the MSRP corpus.

```
Sentence 1: No dates have been set for the civil or the criminal trial.
Sentence 2: No dates have been set for the criminal or civil cases,
but Shanley has pleaded not guilty.

Sentence 1: "His progress is steady, he's stable, he's comfortable,"
Jack said Tuesday afternoon.
Sentence 2: "His progress is steady, he is stable," said Dr Jack.
```

Figure 4.2: Examples of non paraphrases from the MSRP corpus.

We take as train/test set the usual suggested partitions, and further extract a validation set from the train partition, as defined in the GLUE benchmark [Wang et al., 2018]. Hence, we employ 3668 examples for train, 1725 for test, and 408 for validation. The distribution of examples per class is not balanced, namely there are 1147 paraphrases in test, and 2753 in the combination of train and validation sets.

Sentences Involving Compositional Knowledge

Evaluations of NLI and STS are based on the SICK corpus [Marelli et al., 2014b], previously introduced in Section 2.3. We follow the original partitions of 4906 test examples, 495 validation examples and 4439 train examples.

Each example contains two sentences, a continuous value between 1 and 5, and a label with three possible values. For the STS task, the continuous value describes the similarity between the sentences, as previously described in Section 2.1. For the NLI task, the label indicates the relationship between the two sentences, as neutral, contradiction or entailment. Examples of neutral labels are shown in Figure 4.3, while entailment labels are shown in Figure 4.4, and contradiction labels are shown in Figure 4.5, where each Figure contains two examples and the respective similarity scores, and the examples were chosen such that their similarity scores are distant. Most examples of contradiction have large similarity scores, but in Figure 4.5 we include the example with the lowest similarity score in the corpus.

The distribution of labels in the SICK corpus is not balanced, with 5595 examples of neutral, 2821 of entailment and 1424 of contradiction. The distribution of labels per partition is also not balanced. For instance, the train partition contains 2524 neutral examples, 1274 entailment examples, and 641 examples of contradiction. The distribution of labels for the test partition is included in the following NLI results section, as part of the performance evaluation for each label.

```
Sentence 1: A man is jumping into an empty pool
Sentence 2: There is no biker jumping in the air
Similarity: 1.2

Sentence 1: Two children are lying in the snow and are making snow angels
Sentence 1: Two people wearing snowsuits are on the ground making snow angels
Similarity: 4.6
```

Figure 4.3: Examples from the SICK corpus for the neutral label.

```
Sentence 1: A guy is cheerfully playing with a footbag
Sentence 2: The man isn't playing the piano
Similarity: 1.5

Sentence 1: A skilled person is riding a bicycle on one wheel
Sentence 1: A person is riding the bicycle on one wheel
Similarity: 4.3
```

Figure 4.4: Examples from the SICK corpus for the entailment label.

```
Sentence 1: There is no man holding a frog
Sentence 2: A man is holding a frog
Similarity: 2.1

Sentence 1: A band is not performing on a stage
Sentence 1: A band is performing on a stage
Similarity: 4.6
```

Figure 4.5: Examples from the SICK corpus for the contradiction label.

4.1.5 Evaluation Metrics

The metrics employed to measure the performance of our models are described in the following, both for the classification tasks of PI and NLI, and for the the regression task of STS. These metrics follow other systems evaluated on our target corpora, with which we compare the performance of our models. For the computation of all metrics we rely on the *scikit-learn* toolkit [Pedregosa et al., 2011].

For the classification tasks of PI and NLI, we report the evaluation metrics of accuracy, precision, recall and F-score, which produce values in the 0 to 1 range. For the model with best overall performance, we also report these metrics per class. Typically, the computation of such metrics is based on the confusion matrix of a model, where rows and columns are indexed by the classes of the task, with one axis for predicted classes and the other for actual classes, and cells contain the amount of correct and incorrect predictions

relative to a certain class [Sokolova and Lapalme, 2009]. For instance, the first row and column map to the same class, and the corresponding cell contains the amount of examples which were predicted and actually belong to such class.

Most metrics for classification tasks require a problem represented in terms of a positive and a negative class. In a binary classification task, the two involved classes inherently represent the positive and negative case, since one of the classes is the negation of the other. For instance, in PI with the MSRP corpus, an example is either a paraphrase or a non paraphrase. In a multi-class classification task, such as NLI, obtaining the positive and negative classes is typically achieved by addressing the multi-class problem as a set of binary problems [Felkin, 2007]. Namely, the multi-class confusion matrix is transformed into multiple binary confusion matrices, one for each class, where a target class represents the positive class and the set of all remaining classes represents the negative class. To obtain the final score for a metric, we compute a macro average of its results on each confusion matrix, following other systems, which corresponds to an unweighted average where all classes are considered equally important.

A confusion matrix contains the amount of true and false instances of the positive and negative classes, and the computation of classification metrics is based on such amounts. True and false positives and negatives are described in the following, considering a target class as the positive class and the set of the remaining classes as the negative class, as computed for the binary confusion matrices in a multi-class setting. Such description is a generalization of the binary case where the set of remaining/negative classes would only contain one element:

- true positives (TP), were predicted and actually belong to the target class,
- false positives (FP), were predicted as belonging to the target class, but actually belong to one of the remaining classes,
- true negatives (TN), were predicted and actually belong to one of the remaining classes,
- false negatives (FN), were predicted as belonging to one of the remaining classes, but actually belong to the target class,

The sum of TP and FN comprehends all actual examples from a target class, and the sum of TN and FP comprehends all actual examples from the remaining classes. The elements in the diagonal of a confusion matrix are the correct predictions of each class, regardless of the number of classes represented.

Formally, accuracy measures the fraction of correct predictions with respect to all predictions, as shown in Equation (4.1). From the confusion matrix, accuracy is the result of dividing the sum of all diagonal elements by the sum of all elements in the matrix.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

When given the predictions of a model and the actual classes for the examples of such predictions, accuracy also corresponds to the Jaccard similarity score for these two sets, as previously defined in Section 3.1, and this is the computation employed by *scikit-learn*, in which we rely to compute the global accuracy of a model. However, for per class accuracy, we manually compute accuracy from the confusion matrix, using the above equation.

When computing accuracy per class in a binary classification task, accuracy is identical for both classes, and identical to global accuracy, since the diagonal of the confusion matrix for each class contains the same

elements, in different order. In a multi-class classification task, accuracy is different for each class, since the negative classes are different for each target class.

In imbalanced corpora, with more examples of one class than of others, accuracy is not reliable for performance evaluation, since, for instance, classifying all examples as belonging to the class of the majority of examples would produce a competitive result, even though the minority class is disregarded [Chawla, 2005].

Precision measures the performance of a model in correctly predicting a certain class, of all its predictions for such class. It is implemented as the number of examples where the predicted class matches the actual class, divided by the total number of examples predicted as being of said class, as shown in Equation (4.2).

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Recall measures the fraction of examples of a certain class that were correctly predicted, and is implemented as the number of examples wherein the predicted class matches the actual class, divided by the total number of examples of said class, as shown in Equation (4.3).

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

Recall penalizes a model that misclassifies examples of the target class as belonging to the remaining classes (low chance of detecting positives), while precision penalizes a model that misclassifies examples of the remaining classes as belonging to the target class (low confidence on predictions).

The F score combines precision and recall, using an importance factor of precision in respect to recall, as shown in Equation (4.4). We consider precision and recall to have the same weight/importance, as such we define $\beta = 1$ and the metric becomes F1.

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall} \tag{4.4}$$

For STS, as in the SemEval edition that employs the SICK corpus [Marelli et al., 2014a], we report Pearson and Spearman correlations, and the Mean Squared Error (MSE), all of which are suitable to measure the performance of a system that outputs a single unbounded and real valued prediction. A lower MSE is better, while for the remaining metrics a greater value is better.

To compute the MSE, the difference between each prediction and its true value is squared, so that all differences are positive numbers, and the average of all such values is the MSE. Hence, the lowest possible MSE value is 0 and there is no upper bound.

The Pearson correlation coefficient measures the strength and direction of the linear relation between predictions and true values, and corresponds to a continuous value from -1 to 1, where 0 indicates no linear relationship. Visually, a prediction and its true value is represented as a point in a bi-dimensional space, and there is a linear relationship between predictions and true values if all points are near a single line/path. The sign of the Pearson coefficient is the same as the slope for such line, and its value indicates the proximity of points to the line. For instance, a Pearson value near -1 indicates that predictions and true values have distant magnitudes, but vary proportionally on most examples.

The Spearman correlation coefficient is defined as Pearson but instead considers predictions and true values as ranks, and not their actual values. Namely, the Spearman coefficient corresponds to the Pearson applied

to such ranks.

4.2 Results

The performance of our models in addressing the tasks of PI, NLI and STS is reported in the following, relative to English corpora. Namely, for PI our models are built with supervision on the train set of the MSRP corpus, tuned on its validation set, and evaluated on its test set, according to the previously mentioned definition of such partitions. NLI and STS are also addressed with such procedure, but instead using the SICK corpus for both tasks. We present three tables of results for each task, corresponding to the performances of other systems, of our models based only on lexical and/or DRS features, and of our models based also on BERT embeddings. The best results of each metric on each of these tables are highlighted with boldface.

Results for other systems were obtained from the original publications, and some do not report all the evaluation metrics we report for our models. Moreover, some of the other systems employ additional data or combine multiple corpora, while our models are trained per corpus and only with the data in such corpus. The first three reported other systems in NLI and STS tasks are from the SemEval challenge, previously described in Section 2.2, where the SICK corpus was first introduced. In STS, these three results were obtained from the main task publication [Marelli et al., 2014a], since the original publications of each system did not report all evaluation metrics. However, we identify the systems by their original publications.

Results for our models are organized according to the type of model, distinguishing results from fine tuned models and results from the various combinations of features. Models are identified by the corresponding learning algorithm, and models also based on BERT are further identified by the type of BERT model (Base or Large). Learning algorithms are abbreviated as *lin* for linear SVM, *rf* for random forests, *poly* and *rbf* for SVM with polynomial and RBF kernels respectively, and *voting* for the ensemble of all these and also decision trees (all non-deep-learning models).

For models based on feature vectors, we present only the two best results of each combination of features, although models with identical performance are presented together. The remaining of such models are considered non competitive, although mentioned in the presentation of results. For fine tuned models, we report the mean and standard deviation of scores from five instances of each model, for each metric.

For the classification tasks of PI and NLI, we also report per class results for an ensemble of five instances of the fine tuned BERT-Large model, obtained by averaging their predictions, since this model achieves the best results on most tasks and evaluation metrics. With the per class results, we also mention the amount of examples from each class, as a fraction of the total examples in the corpus.

4.2.1 Paraphrase Identification

A selection of results for the PI task on the MSRP corpus is shown in the following tables. Namely, Table 4.1 shows the performance of other systems, both from early and modern works, Table 4.2 shows the results from our models based on lexical and/or semantic features, Table 4.3 is for our models where BERT embeddings are involved, and finally Table 4.4 shows the per class results of the fine tuned BERT-Large model.

Early works shown in Table 4.1 correspond to systems based on lexical distance metrics [Madnani et al., 2012] and tree/graph kernels [Filice et al., 2015], but other methods are available in the ranking for the MSRP corpus, previously mentioned in Section 2.3. Various modern works report results on this task and

corpus as part of their participation in the GLUE benchmark, previously described in Section 2.2. Some of these are shown in Table 4.1 and leverage neural networks [Liu et al., 2019c] and/or BERT embeddings, both from original [Devlin et al., 2019] and modified versions [Wang et al., 2020] or BERT. The latter corresponds to the best accuracy and F1 scores in our selection of other systems, and is also one of the best scores in the GLUE benchmark, although its method requires costly computing resources. Instead, the original BERT model is more cost accessible and still provides competitive performance from modern embeddings technology.

Table 4.1: Results from other systems, for the PI task on the MSRP corpus.

System	Accuracy	F1
[Madnani et al., 2012]	0.77	0.84
[Filice et al., 2015]	0.79	0.85
[Camburu et al., 2018b]	0.76	0.83
[Wang et al., 2019b]	0.75	0.82
[Devlin et al., 2019]	0.85	0.89
[Liu et al., 2019c]	0.88	0.91
[Yang et al., 2019b]	0.75	0.83
[Wang et al., 2020]	**0.92**	**0.94**
[Ahmed and Mercer, 2020]	0.76	

Regarding our models based on lexical and/or DRS features, partially shown in Table 4.2, the best results are achieved by voting models or SVM models of any kernel. These achieve better performance than random forests, by a difference of at most 0.02 in either accuracy or F1. Decision trees are competitive with any other model. Particularly, models based only on DRS features achieve better results with decision trees than with random forests, in at most 0.01 of either accuracy or F1, and results with decision trees are similar to those of SVM with RBF kernel. Using only DRS features results in lower performance on most evaluation metrics than using only lexical features, and the combination of both features sets produces results similar to those of using lexical features only. As such, when combining lexical and DRS features on this corpus and task, DRS features have no impact on the performance of lexical features.

Table 4.2: Results for the PI task on the MSRP corpus, without considering BERT.

System	Accuracy	Precision	Recall	F1
# LEX				
poly	**0.78**	**0.82**	0.85	0.83
voting	**0.78**	**0.82**	0.86	**0.84**
# DRS				
poly	0.73	0.75	**0.90**	0.81
voting	0.73	0.75	0.89	0.82
# LEX + DRS				
lin	**0.78**	**0.82**	0.86	**0.84**
rbf	**0.78**	**0.82**	0.85	**0.84**

For models based on feature vectors that contain BERT embeddings, shown in Table 4.3, results are similar with both Base and Large versions of BERT. Voting models achieve the best scores on all evaluation metrics, followed by models based on the various SVM kernels, which differ between each other in at most 0.01 of either accuracy or F1. Results with random forests are competitive, although inferior to any SVM, and decision trees are not competitive, with at most less 0.05 than random forests, in either accuracy or F1.

The per class results for our best performing model, the fine tuned BERT-Large model, are shown in Table

Table 4.3: Results for the PI task on the MSRP corpus, involving BERT embeddings.

System	Accuracy	Precision	Recall	F1
# BERT fine tuned				
BERT-Large	**0.83 ± 0.02**	**0.85 ± 0.05**	**0.90 ± 0.05**	**0.87 ± 0.01**
BERT-Base	0.82 ± 0.01	0.84 ± 0.03	0.89 ± 0.03	**0.87 ± 0.00**
# BERT as features				
BERT-Large (lin)	0.75	0.79	0.85	0.82
BERT-Large (voting)	0.76	0.80	0.86	0.82
BERT-Base (rbf)	0.75	0.80	0.83	0.81
BERT-Base (voting)	0.76	0.78	0.88	0.83
# BERT as features + LEX				
BERT-Large (lin)	0.79	0.83	0.85	0.84
BERT-Large (voting)	0.78	0.82	0.86	0.84
BERT-Base (poly)	0.78	0.83	0.83	0.83
BERT-Base (voting)	0.78	0.82	0.86	0.84
# BERT as features + DRS				
BERT-Large (lin)	0.76	0.80	0.86	0.83
BERT-Large (voting)	0.77	0.80	0.87	0.83
BERT-Base (rbf)	0.75	0.81	0.82	0.81
BERT-Base (voting)	0.75	0.79	0.86	0.82
# BERT as features + LEX + DRS				
BERT-Large (lin)	0.78	0.83	0.85	0.84
BERT-Large (rbf)	0.79	0.83	0.85	0.84
BERT-Base (rbf)	0.78	0.83	0.85	0.84
BERT-Base (voting)	0.78	0.82	0.87	0.84

4.4 and indicate a better performance in detecting paraphrases. However, given that the MSRP corpus is imbalanced and contains more paraphrases than non paraphrases, both in train and test, the performance in detecting non paraphrases is considered competitive. Since PI is a binary classification task, the per class accuracy is identical in both classes. As previously mentioned, the per class results are obtained from an ensemble of the five instances of the fine tuned BERT-Large model, as such the accuracy is different than reported in Table 4.3 for the mean of the five instances.

Table 4.4: PI results on MSRP, per class and relative to the fine tuned BERT-Large model.

Label	Accuracy	Precision	Recall	F1	Examples from Total
NOT paraphrase	0.85	0.82	0.72	0.77	0.34
paraphrase	0.85	0.87	0.92	0.89	0.66

4.2.2 Natural Language Inference

A selection of results for the NLI task on the SICK corpus is shown in the following tables, namely in Table 4.5 for other systems, both from early and modern works, in Table 4.6 for our models based on lexical and semantic features, in Table 4.7 for our models based also on BERT embeddings, and in Table 4.8 for the per class results of the fine tuned BERT-Large model.

One of the other systems reported in Table 4.5 also employs DRS to compute features, using similar tools as we employed, although the only feature computed directly from DRS is based on the overlap between

some semantic roles [Bjerva et al., 2014]. Most modern works also report results in the PI task, but none employs BERT. The best score from other systems is obtained by three systems, which report identical accuracy scores, although from distinct approaches.

Table 4.5: Results from other systems, for the NLI task on the SICK corpus.

System	Accuracy
# Other systems	
[Lai and Hockenmaier, 2014]	0.85
[Zhao et al., 2014]	0.84
[Bjerva et al., 2014]	0.82
[Wang et al., 2019b]	0.83
[Camburu et al., 2018b] / [Yang et al., 2019b] / [Ahmed and Mercer, 2020]	**0.86**

All of our models based only on DRS features achieve similar results, and the largest performance difference between them is 0.03, in any metric, which occurs when comparing the results from linear SVM (the best result) and decision trees. For models based only on lexical features, random forests and SVM with polynomial kernel achieve identical results, while decision trees are the only non competitive model, being at most 0.14 worse than the best model (obtained by voting). Unlike with models based only on DRS features, with lexical features linear SVM produces the second worst result, and all remaining models achieve similar results. For models that combine lexical and DRS features, results follow the same variations as in models based only in lexical features, except for linear SVM which now achieves competitive performance. DRS features contribute to increase the performance of lexical features alone, and the combination of the two feature sets produces one of the best overall results, even when considering the non fine tuned BERT based models described in the following.

Table 4.6: Results for the NLI task on the SICK corpus, without considering BERT.

System	Accuracy	Precision	Recall	F1
# LEX				
rf / poly	0.81	0.83	0.75	0.78
voting	0.82	0.84	0.76	0.79
# DRS				
lin	0.79	0.79	0.77	0.78
poly / voting	0.78	0.80	0.75	0.77
# LEX + DRS				
rf	**0.84**	**0.86**	**0.80**	**0.82**
voting	0.83	**0.86**	0.79	**0.82**

Using only BERT embeddings as features, all models achieve similar performance, both with BERT-Base and BERT-Large, except models relying on decision trees (random forests and decision trees), which perform worse than all remaining models in at least 0.06 for any metric. All models improve performance in most metrics when considering BERT embeddings combined with lexical or DRS features, but for decision trees the improvement is more noticeable, achieving at most 0.17 more in some metrics when compared to using BERT alone. In such combinations, results with either lexical or DRS features are similar or identical, with the greatest difference occurring in the BERT-Base decision trees model, where combining BERT with DRS features increases recall in 0.05 more than achieved with lexical features.

For all models involving BERT embeddings, the best performance is achieved when combining BERT embeddings with both lexical and DRS features, except for models based on decision trees, where the best performance is achieved when combining BERT with DRS features, in both BERT-Base and BERT-Large.

Models based on random forests and decision trees achieve better performance with BERT-Base, while the remaining models achieve better performance with BERT-Large, by a difference of at most 0.02 in some metrics.

Table 4.7: Results for the NLI task on the SICK corpus, involving BERT embeddings.

System	Accuracy	Precision	Recall	F1
# BERT fine tuned				
BERT-Large	**0.89 ± 0.01**	**0.88 ± 0.01**	**0.88 ± 0.00**	**0.88 ± 0.01**
BERT-Base	0.87 ± 0.00	0.86 ± 0.00	**0.88 ± 0.00**	0.87 ± 0.00
# BERT as features				
BERT-Large (lin / poly / voting)	0.83	0.84	0.80	0.82
BERT-Large (rbf)	0.82	0.82	0.80	0.81
BERT-Base (lin / poly / rbf)	0.81	0.81	0.79	0.80
BERT-Base (voting)	0.80	0.81	0.78	0.79
# BERT as features + LEX				
BERT-Large (lin)	0.84	0.85	0.82	0.83
BERT-Large (poly)	0.84	0.86	0.81	0.83
BERT-Base (lin)	0.83	0.83	0.81	0.82
BERT-Base (poly / voting)	0.83	0.85	0.80	0.82
# BERT as features + DRS				
BERT-Large (lin)	0.84	0.85	0.82	0.83
BERT-Large (poly / voting)	0.84	0.85	0.81	0.83
BERT-Base (lin / rbf)	0.81	0.82	0.80	0.81
BERT-Base (poly / voting)	0.82	0.83	0.80	0.81
# BERT as features + LEX + DRS				
BERT-Large (lin)	0.85	0.85	0.83	0.84
BERT-Large (poly / voting)	0.85	0.87	0.82	0.84
BERT-Base (lin)	0.83	0.83	0.81	0.82
BERT-Base (voting)	0.84	0.85	0.81	0.83

The per class results shown in Table 4.8 indicate competitive performance in detecting all classes, despite the imbalance in the SICK corpus. For instance, 57% of the examples in this corpus belong to the neutral class, while only 15% are contradictions, still our fine tuned BERT-Large model achieves similar F1 scores for both.

As previously mentioned in Section 4.1.5, accuracy is not a valid indicator of per class performance for infrequent classes. For instance, contradiction is the class with fewer examples in SICK, but even if all contradictions are misclassified, the accuracy for the contradiction class would be erroneously competitive, since the diagonal of the confusion matrix for contradictions would still have the majority of examples, corresponding to examples classified as contradictions in true positives and examples classified as any other class in true negatives.

Table 4.8: NLI results on SICK, per class and relative to the fine tuned BERT-Large model.

Label	Accuracy	Precision	Recall	F1	Examples from Total
neutral	0.90	0.92	0.90	0.91	0.57
entailment	0.93	0.84	0.91	0.87	0.29
contradiction	0.97	0.92	0.85	0.88	0.14

4.2.3 Semantic Textual Similarity

A selection of results for the STS task on the SICK corpus is shown in the following tables, namely in Table 4.9 for other systems, both from early and modern works, in Table 4.10 for models we produced based on lexical and semantic features, and in Table 4.11 for models we produced based on BERT embeddings.

The other system based on DRS, previously mentioned in the NLI results, achieves the best MSE of our selection of other systems, as shown in Table 4.9, and also the best MSE of all systems in the original shared task where the SICK corpus was introduced [Marelli et al., 2014a]. Modern systems do not report MSE, but achieve the best overall Pearson scores, particularly by leveraging additional data [Camburu et al., 2018b] or dependency parse trees [Liu et al., 2019a] to produce sentence representations. However, the results of these systems are similar to our best results, obtained with the BERT sentence representations which involve less complexity and do not require specialized resources.

Table 4.9: Results from other systems, for the STS task on the SICK corpus.

System	MSE	Pearson	Spearman
# Other systems			
[Zhao et al., 2014]	0.33	0.83	0.77
[Bjerva et al., 2014]	**0.32**	0.83	0.78
[Jimenez et al., 2014]	0.36	0.80	0.75
[Camburu et al., 2018b]		**0.89**	
[Wang et al., 2019b]		0.86	
[Liu et al., 2019a]		**0.89**	**0.83**
[Yang et al., 2019b]		0.87	

Regarding our models based only on lexical features, decision trees and linear SVM achieve similar results and the worst performance, while SVM with the remaining non linear kernels also achieve similar results, but with competitive performance. For models based only on DRS features, random forests and SVM with non linear kernels achieve similar and competitive results, while decision trees achieve the worst performance by a difference of at most 0.09 in MSE, 0.05 in Pearson and 0.07 in Spearman. For the combination of lexical and DRS features, random forests achieve better performance than non linear SVM, particularly in MSE by at most 0.03, and the best result of all models when not considering BERT embeddings. As in the results for the NLI task, models using DRS features combined with lexical features achieve better performance than those using only lexical or DRS features alone.

Table 4.10: Results for the STS task on the SICK corpus, without considering BERT.

System	MSE	Pearson	Spearman
# LEX			
rf	0.37	0.80	0.74
voting	0.39	0.79	0.73
# DRS			
rbf	0.46	0.74	0.70
voting	0.44	0.75	0.71
# LEX + DRS			
rf	**0.34**	**0.82**	**0.76**
voting	0.35	0.81	**0.76**

Using only BERT embeddings as features, all SVM based models and the voting model achieve similar results, and the best performances. The same occurs when combining BERT embeddings with lexical

and/or DRS features, but random forests are now also competitive, achieving similar performance to linear SVM. Models based on decision trees achieve the worst performance in all feature combinations. Excluding decision trees, the difference between the best and worst performances is at most 0.06 on any metric, with the greatest differences occurring when using only BERT embeddings. Using BERT-Base or BERT-Large produces similar results for all models, thus no significant advantage is achieved by increasing the complexity in BERT models, although BERT-Large models perform better in all experiments. Only the combination of BERT with lexical and DRS features is able to match the best overall results (and only in MSE), obtained with the fine tuned BERT model.

Table 4.11: Results for the STS task on the SICK corpus, involving BERT embeddings.

System	MSE	Pearson	Spearman
# BERT fine tuned			
BERT-Large	**0.26 ± 0.02**	**0.88 ± 0.01**	**0.84 ± 0.00**
BERT-Base	0.30 ± 0.00	**0.88 ± 0.00**	0.83 ± 0.00
# BERT as features			
BERT-Large (poly)	0.30	0.84	0.78
BERT-Large (rbf)	0.29	0.85	0.79
BERT-Base (poly)	0.31	0.83	0.76
BERT-Base (rbf)	0.31	0.83	0.76
# BERT as features + LEX			
BERT-Large (rbf)	0.27	0.86	0.80
BERT-Large (voting)	0.28	0.85	0.80
BERT-Base (poly)	0.28	0.85	0.79
BERT-Base (rbf)	0.29	0.85	0.78
# BERT as features + DRS			
BERT-Large (rbf)	0.28	0.85	0.80
BERT-Large (voting)	0.28	0.86	0.80
BERT-Base (poly)	0.30	0.84	0.78
BERT-Base (voting)	0.30	0.84	0.77
# BERT as features + LEX + DRS			
BERT-Large (rbf)	**0.26**	0.86	0.81
BERT-Large (voting)	0.27	0.86	0.80
BERT-Base (rbf)	0.28	0.85	0.79
BERT-Base (voting)	0.28	0.85	0.79

4.3 Discussion

In the tasks of NLI and STS, models using DRS features combined with lexical features achieved better results than with lexical features only, while on the task of PI, combining lexical and DRS features did not alter the performance relative to using lexical features only. Since the texts in the MSRP corpus, that supports the PI evaluation, are longer and more complex than the texts in the SICK corpus, which supports both NLI and STS, it is expected that the DRS produced from MSRP texts are more prone to errors than those from SICK texts. Also, the WordNet based word expansion employed by DRS features may further increase the impact of such errors in the final feature set representation, even after robust feature scaling as employed.

Some of the other systems report results in all tasks, using techniques such as swapping the sentences in a percentage of the pairs [Wang et al., 2019b], leveraging human explanations of entailment [Camburu

et al., 2018b], or modelling the semantic novelty of words [Yang et al., 2019b].

While fine tuning and training with more data are popular techniques, some of the best results from our reported other systems are obtained by using only the original data and focusing on modelling aspects, as also employed in our feature based models. For instance, two of the best results we report for other systems in NLI employ only original corpora and focus on the contribution of words to the overall semantics in a pair, by measuring and modelling their uniqueness [Yang et al., 2019b] or irrelevance [Ahmed and Mercer, 2020], while using advanced modelling techniques. However, for the corpora and tasks in our evaluation, the results achieved by these systems are similar to those achieved by our models, based on traditional machine learning methods applied to BERT embeddings combined with lexical and DRS features.

In fine tuned models, we measured statistical significance between the values returned by the pair of models, using a t-test (five runs for each model) for all metrics. Considering $p = 0.05$, there are statistically significant differences between models BERT-Base and BERT-Large for all evaluation metrics and tasks, except for the Pearson metric in STS and all evaluation metrics in PI except F1.

Fine tuned models achieved better results than traditional models in most of the evaluated corpora and tasks. Particularly, the BERT-Large fine tuned model achieved the best results in all corpora and tasks, compared to our other setups and to most other systems. However, for instance, in PI it achieved the same recall as the model based only on DRS features, and in STS it achieves the same MSE as the model combining BERT embeddings, lexical features and DRS features. Compared to BERT-Base, it achieves the same Pearson in STS, the same recall in NLI and the same F1 in PI.

Regarding classification with models involving DRS features, some examples were only correctly classified by models based only on DRS features. For instance, regarding the PI task and the test set of the MSRP corpus, the best model based only in DRS features, which corresponds to an SVM with polynomial kernel, according to the previously shown results, correctly classified the examples in Figures 4.6, for the paraphrase class, and 4.7, for the non paraphrase class, while all the best models combining DRS features with BERT embeddings and/or lexical features misclassified these examples. A total of 34 of the 1725 examples in the test set of the MSRP corpus were only correctly classified by the model based only in DRS features.

```
Sentence 1: Veteran entertainer Bob Hope celebrates his 100th birthday -
    and many years in showbusiness - on Thursday.
Sentence 2: Hollywood and the world are gearing up to celebrate legendary
    entertainer Bob Hope's 100th birthday on Thursday.

Sentence 1: The Prime Minister, Junichiro Koizumi, joined the criticism.
Sentence 2: Prime Minister Junichiro Koizumi said Mr Ota deserved to be criticised.
```

Figure 4.6: Examples of paraphrases from the test set for the PI task, which were only correctly classified by the best model based only on DRS features, among all models that consider DRS features.

In contrast, and again only considering models that involve DRS features, some examples were only misclassified by the model based only on DRS features, while all the remaining models based on DRS features correctly classified such examples. For instance, regarding the PI task, the best models combining DRS features with BERT embeddings and/or lexical features correctly classified the examples in Figures 4.8, for paraphrases, and 4.9, for non paraphrases, while the best model based only in DRS features misclassified these examples. A total of 104 of the 1725 examples in the test set of the MSRP corpus were only misclassified by the model based only in DRS features.

Regarding the other classification task of NLI, and since the PI examples from the MSRP corpus are too

```
Sentence 1: Penn Traffic's stock closed at 36 cents per share on Wednesday
    on Nasdaq, up two cents.
Sentence 2: Penn Traffic stock closed Wednesday at 36 cents, up 2 cents,
    or 6.2 percent, from Tuesday's close.

Sentence 1: Of 24 million phoned-in votes, 50.28 percent were for Studdard,
    putting him 130,000 votes ahead of Aiken.
Sentence 2: Of the 24 million phone votes cast, Studdard was only 130,000
    votes ahead of Aiken.
```

Figure 4.7: Examples of non paraphrases from the test set for the PI task, which were only correctly classified by the best model based only on DRS features, among all models that consider DRS features.

```
Sentence 1: Dynes will get $395,000 a year, up from Atkinson's
    current salary of $361,400.
Sentence 2: In his new position, Dynes will earn $395,000, a
    significant increase over Atkinson's salary of $361,400.

Sentence 1: Along with chipmaker Intel, the companies include
    Sony Corp., Microsoft Corp., Hewlett-Packard Co., IBM Corp.,
    Gateway Inc. and Nokia Corp.
Sentence 2: Along with chip maker Intel, the companies include
    Sony, Microsoft, Hewlett-Packard, International Business Machines,
    Gateway, Nokia and others.
```

Figure 4.8: Examples of paraphrases from the test set for the PI task, which were correctly classified by all models involving DRS features, except the model based only on DRS features.

complex for analysis, in Figure 4.10 we present examples from the test set of the SICK corpus that were only correctly classified by the best model based only on DRS features, which for NLI corresponds to a SVM with linear kernel, according to the previously shown results. Examples that were correctly classified by all models involving DRS features, except the model based only on DRS features, are shown in Figure 4.11. For NLI, a total of 41 of the 4906 examples in the test set of SICK was only correctly classified by the model based only on DRS features, and a total of 265 of the 4906 examples was only misclassified by such model.

```
Sentence 1: Doud was shot in the shoulder and underwent surgery at Strong
    Memorial Hospital, where he was listed in satisfactory condition.
Sentence 2: A spokeswoman at Strong Memorial Hospital said Doud was in
    satisfactory condition Tuesday night.

Sentence 1: At 11:30 a.m., Edmund Hillary of New Zealand and Tenzing
    Norgay Sherpa of Nepal reached the summit.
Sentence 2: Sherpa Tenzing Norgay, who reached the summit with Sir
    Edmund, died in 1986.
```

Figure 4.9: Examples of non paraphrases from the test set for the PI task, which were correctly classified by all models involving DRS features, except the model based only on DRS features.

```
Sentence 1: A soccer player is sitting on the field and is drinking water
Sentence 2: Water is being drunk by a soccer player sitting on the field
Class: Entailment

Sentence 1: A deer is jumping over the enclosure
Sentence 2: A deer is jumping a fence
Class: Neutral

Sentence 1: A man is pointing at a silver sedan
Sentence 2: There is no man pointing at a car
Class: Contradiction
```

Figure 4.10: Examples from the test set for the NLI task, which were only correctly classified by the model based only on DRS features, among all models that consider DRS features.

```
Sentence 1: A boy is playing slip and slide in the grass
Sentence 2: A child is playing slip and slide in the grass
Class: Entailment

Sentence 1: Children in red shirts are playing in the leaves
Sentence 2: Children covered by leaves are playing with red shirts
Class: Neutral

Sentence 1: A lot of people are in an ice skating park
Sentence 2: There aren't many people in the ice skating park
Class: Contradiction
```

Figure 4.11: Examples from the test set for the NLI task, which were correctly classified by all models involving DRS features, except the model based only on DRS features.

5

Evaluation on the Impact of Semantic Features for Portuguese

To assess the similarity between Portuguese sentences, we designed models that leverage BERT embeddings for Portuguese only, multilingual BERT embeddings, and lexical similarity metrics, separately and combined. In the following sections we describe the experimental setup to evaluate such models, and discuss the corresponding results. Namely, we evaluate models supervised on corpora suitable for the tasks of PI, NLI and STS, employing the corresponding test sets, and our models include fine tuning BERT embeddings on such corpora, using generic BERT embeddings in traditional learning algorithms, such as SVM, and also combining such embeddings with lexical features. For this experiment, we consider BERT embeddings as semantic features.

5.1 Experimental Setup

In this section, we describe the corpora where the performance of our models is assessed, and the configurations employed to compute language dependent features and embeddings. Model configuration and evaluation metrics are the same as employed for English.

5.1.1 Lexical Similarity Features

The language dependent lexical features of our system were adapted to Portuguese as described in the following.

To compose word clusters, previously described in Section 3.1, the Brown clustering algorithm [Brown et al., 1992] is applied to a collection of news documents from the Portuguese newspaper *Público*, as obtained from [Marques, 2015].

Negative words, required by the NEG Overlap feature previously described in Section 3.1, are defined as *não*, *nunca*, *jamais*, *nada*, *nenhum*, *ninguém*, following [Marques, 2015].

Modal words, required by the MODAL Overlap feature previously described in Section 3.1, are defined as *podia*, *poderia*, *dever*, *deve*, *devia*, *deverá*, *deveria*, *faria*, *possivel*, *possibilidade*, *possa*, following [Marques, 2015].

5.1.2 BERT Embeddings

As in the English experiments, we obtained embeddings from generic BERT models, as a single embedding for the concatenation of two target sentences. We employed the base and large versions of the Portuguese BERT model (ptBERT-Base and ptBERT-Large) described in `https://github.com/neuralmind-ai/portuguese-bert`, which was pre-trained in Brazilian corpora [Souza et al., 2020]. We also employed the multilingual BERT model, which is only available in base version (mBERT-Base) and was pre-trained on Wikipedia [Pires et al., 2019]. As in the English experiments, we always employ raw text that was not transformed by lower casing or accent removal.

5.1.3 Corpora

ASSIN

The ASSIN dataset [Fonseca et al., 2016] contains 10000 sentence pairs collected from Google News, split into training and test sets with an equal number of Portuguese and Brazilian examples in each set (each Portuguese variety has 2500 examples for training, 500 for trial and 2000 to test). Following the ASSIN campaign, in our experiments performance was measured separately for European Portuguese and Brazilian Portuguese, but also for the concatenation of both corpora. These partitions are henceforth mentioned as ASSIN-PTPT, ASSIN-PTBR and ASSIN, respectively.

The vocabulary employed in the Brazilian split is partially different than that of the Portuguese split. For instance, the Brazilian split includes sentence "O time estreia na Copa América contra o Peru.", where the word "time", which means team in English, is specific to the Brazilian vocabulary, since "time" does not exist in Portuguese, where the word for team would be "equipa".

Each example is annotated for both the STS and RTE tasks. For STS, semantic relatedness is a continuous value from 1 to 5, according to the guidelines [Fonseca et al., 2016], previously mentioned in 2.1. RTE is defined as a categorical assignment to the classes entailment, paraphrase or none (neutral), which we here consider as the task of PI. The distribution of examples with these labels is approximately balanced between the European and Brazilian splits of this corpus but is not balanced relative to examples per label, with a total in both corpora of 7316 examples of neutral, 2080 of entailment and 604 of paraphrase.

SICK-BR

The SICK-BR corpus [Real et al., 2018] is a Brazilian Portuguese translation of the SICK [Marelli et al., 2014b] corpus (NLI and STS). As in the original SICK corpus, SICK-BR is composed of 4906 test examples, 495 trial/development examples and 4439 train examples.

Each example, that is, each sentence pair, is annotated for the STS task with a continuous value between 0 and 5, for how similar the two sentences are; and for the NLI task, with labels neutral, contradiction and entailment to indicate the relationship between the two sentences. The distribution of these labels in the corpus is not balanced, with 5595 examples of neutral, 2821 of entailment and 1424 of contradiction. More information is available, such as the original English sentences. The development of SICK-BR only targeted sentence translation; hence, the remaining annotations are the same as in SICK.

ASSIN2

ASSIN2 [Real et al., 2020] extends SICK-BR with more examples. These examples were created by modifying examples of SICK-BR, by replacing words by their synonyms, for instance. ASSIN2 has approximately the same size as SICK-BR, although it does not include contradiction examples. Instead, the NLI task is defined as a binary classification, for the classes of entailment and not entailment. Example distribution for ASSIN2 contains 6500 examples for training, 500 for validation and 2448 for testing, and this is the only corpus in our evaluation that describes a balanced distribution of examples per label, where exactly half of the examples on each corpus partition are of the entailment class, while examples of the other half correspond to the not entailment class. The STS task is also included in ASSIN2, as inherited from SICK-BR.

5.2 Results

The results of our models in the tasks of PI, NLI and STS, using Portuguese corpora, are reported in the following. For each corpus, we provide a table with results from our models and from other systems, where the best results for each metric are highlighted with boldface.

Results for our models are designed as in the English evaluation, both in organization, abbreviations for learning algorithms and information provided for fine tuned models. For per class results, the Portuguese equivalent to the BERT-Large model (ptBERT-Large) is instead employed. Unlike in the English experiments, we only report two results per BERT model and feature set when particular cases occur, such as when the best accuracy and best F1 scores are achieved by different models, or if the second best result is similar to the best result but was instead achieved by a less computationally expensive model. Nonetheless, we mention results from models that we computed which did not achieved competitive results, in support of each table.

Results for other systems were obtained from the original publications, according to the therein addressed tasks, corpora and evaluation metrics. For each corpus, we report results for all other systems, to the best of our knowledge, that achieve competitive performances. Some systems report results on multiple corpora, and not all of the evaluation metrics we report for our systems are reported in the original publications of other systems. Moreover, some of the other systems were trained with additional data or combine multiple corpora, while our systems are trained per corpus and only with the data in such corpus.

5.2.1 Paraphrase Identification

A selection of results for PI on the ASSIN-PTPT corpus is shown in Table 5.1, where our fine tuned model based on ptBERT-Large embeddings achieved the best overall performance in all metrics. Our model based on lexical features and linear SVM achieved the best performance of all traditional models, similar performance to the fine tuned mBERT-Base model, and better performance than all other systems, except for the F1 score from one of the other systems [Pinheiro et al., 2017]. All our models based only on lexical features achieved similar accuracy, and the only of such models where F1 was not competitive were based on random forests or decision trees.

Table 5.1: PI results on ASSIN-PTPT.

System	Accuracy	Precision	Recall	F1
# Other systems				
[Rocha and Lopes Cardoso, 2018]	0.84			0.73
[Pinheiro et al., 2017]	0.83			0.82
[Barbosa et al., 2016]	0.78			0.61
[Oliveira Alves et al., 2016]	0.79			0.58
# BERT fine tuned				
ptBERT-Large	**0.91 ± 0.00**	**0.87 ± 0.01**	**0.81 ± 0.01**	**0.84 ± 0.00**
ptBERT-Base	0.90 ± 0.01	0.86 ± 0.03	0.76 ± 0.03	0.80 ± 0.02
mBERT-Base	0.87 ± 0.01	0.77 ± 0.03	0.79 ± 0.02	0.76 ± 0.04
# BERT as features				
ptBERT-Large (lin)	0.83	0.69	0.63	0.65
ptBERT-Large (rbf)	0.80	0.65	0.69	0.67
ptBERT-Base (lin)	0.83	0.70	0.66	0.68
mBERT-Base (lin)	0.85	0.71	0.68	0.70
# BERT as features + LEX				
ptBERT-Large (lin)	0.84	0.69	0.64	0.65
ptBERT-Large (dt)	0.78	0.64	0.68	0.64
ptBERT-Base (lin)	0.85	0.72	0.71	0.72
mBERT-Base (lin)	0.85	0.71	0.73	0.72
# LEX				
lin	0.86	0.76	0.74	0.75

When using traditional models with only BERT embeddings as features, linear SVM achieves the best performance in most experiments. With ptBERT-Large, the best accuracy and best F1 scores are obtained by different models, based on SVM with linear and RBF kernels, respectively. In models based on ptBERT-Base and mBERT-Base embeddings, the performance is similar in models based on voting and on SVM of any kernel.

Combining BERT embeddings with lexical features produces identical or better performance than with BERT embeddings only. When using ptBERT-Large embeddings, the best performances were achieved by the least complex models, and the only non competitive models were based on random forests or SVM with RBF kernel, particularly in F1 scores. For instance, the model based on decision trees achieved a better accuracy score than the model based on the polynomial SVM, which is more complex and computationally costly, and a better F1 score than the voting model. With ptBERT-Base and mBERT-Base embeddings, all models achieved similar accuracy scores, and the only models where F1 was not competitive were based on random forests or decision trees.

The per class results for ASSIN-PTPT, shown in Table 5.2, indicate that performance is proportional to the

amount of examples in each class, although competitive for all classes. For instance, the paraphrase class is represented in only 7% of the examples and still achieves an F1 value of 0.73. Accuracy is misleading due to class imbalance, as previously described in Section 4.1.5.

Table 5.2: PI results on ASSIN-PTPT, per class and relative to the fine tuned ptBERT-Large model.

Label	Accuracy	Precision	Recall	F1	Examples from Total
neutral	0.94	0.93	0.98	0.96	0.69
entailment	0.93	0.90	0.81	0.85	0.24
paraphrase	0.97	0.82	0.66	0.73	0.07

A selection of results for PI on the ASSIN-PTBR corpus are shown in Table 5.3, where all other systems also reported results in ASSIN-PTPT. Most F1 scores were lower than in ASSIN-PTPT, both with our models and in other systems. Of all traditional models, the best accuracy was achieved by using only lexical features, with either a linear or polynomial SVM, and the best F1 was achieved with mBERT-Base embeddings combined with lexical features, using the voting strategy. Such results are identical or better than those obtained with the fine tuned model based on mBERT-Base.

Table 5.3: PI results on ASSIN-PTBR.

System	Accuracy	Precision	Recall	F1
# Other systems				
[Barbosa et al., 2016]	0.82			0.52
[Oliveira Alves et al., 2016]	0.82			0.47
[Pinheiro et al., 2017]	0.85			**0.81**
# BERT fine tuned				
ptBERT-Large	**0.90 ± 0.01**	0.82 ± 0.05	0.67 ± 0.05	0.70 ± 0.06
ptBERT-Base	**0.90 ± 0.00**	**0.83 ± 0.01**	**0.70 ± 0.03**	0.75 ± 0.02
mBERT-Base	0.86 ± 0.00	0.59 ± 0.15	0.53 ± 0.02	0.52 ± 0.01
# BERT as features				
ptBERT-Large (lin)	0.84	0.66	0.57	0.59
ptBERT-Base (voting)	0.85	0.66	0.57	0.60
ptBERT-Base (poly)	0.84	0.68	0.61	0.63
mBERT-Base (lin)	0.85	0.66	0.59	0.62
# BERT as features + LEX				
ptBERT-Large (lin)	0.84	0.68	0.62	0.62
ptBERT-Base (dt)	0.82	0.63	0.58	0.60
ptBERT-Base (poly)	0.80	0.60	0.67	0.63
ptBERT-Base (voting)	0.82	0.63	0.66	0.64
mBERT-Base (lin)	0.84	0.64	0.67	0.65
mBERT-Base (voting)	0.85	0.68	0.65	0.66
# LEX				
poly	0.86	0.70	0.60	0.64

In ASSIN-PTBR, all traditional models based only on BERT embeddings performed similarly with any BERT model, except for models based on decision trees and random forests which had lower performances, particularly in the F1 metric. With ptBERT-Base embeddings, the best accuracy and F1 were achieved by different models, based on voting and polynomial SVM respectively.

For the combination of BERT embeddings and lexical features, the performance with ptBERT-Large embeddings was only competitive with linear SVM, by a difference to other models of at least 0.05 in accuracy

and at least 0.02 in F1. With ptBERT-Base and mBERT-Base embeddings, the best model in all metrics was based on voting. However, for such embeddings, models based on a single learning algorithm achieved similar results in accuracy and F1 scores, as shown in Table 5.3. Since the voting model is more costly to compute than a single model, as it encompasses multiple models, we consider single models as the best choice. In particular, for ptBERT-Base embeddings, the model based on decision trees achieves identical accuracy to the voting model, and similar performance on the remaining metrics, but is more efficient to train or predict than any other model. In most models and metrics, combining BERT embeddings with lexical features produces identical or better performance than with BERT embeddings only.

In the per class results for ASSIN-PTBR, shown in Table 5.4, performance is proportional to the amount of examples on each class, but unlike in the evaluation for ASSIN-PTPT, the performance of the least represented class (paraphrase) is not competitive, although the amount of examples in such class is similar as in ASSIN-PTPT.

Table 5.4: PI results on ASSIN-PTBR, per class and relative to the fine tuned ptBERT-Large model.

Label	**Accuracy**	**Precision**	**Recall**	**F1**	**Examples from Total**
neutral	0.94	0.94	0.98	0.96	0.78
entailment	0.91	0.72	0.75	0.73	0.17
paraphrase	0.96	0.95	0.33	0.49	0.05

A selection of results for PI on the full ASSIN corpus is shown in Table 5.5. All traditional models based only on BERT embeddings performed similarly with any BERT model, except for models based on decision trees and random forests which were not competitive, particularly in the F1 metric. All models based only on lexical features achieved competitive results, and similar to the results obtained from models using only BERT as features.

Table 5.5: PI results on ASSIN (PTPT + PTBR).

System	**Accuracy**	**Precision**	**Recall**	**F1**
# Other systems				
[Barbosa et al., 2016]	0.80			0.58
[Oliveira Alves et al., 2016]	0.80			0.54
[Pinheiro et al., 2017]	0.83			**0.83**
# BERT fine tuned				
ptBERT-Large	**0.91 ± 0.01**	**0.84 ± 0.03**	**0.81 ± 0.05**	0.82 ± 0.04
ptBERT-Base	0.90 ± 0.01	0.81 ± 0.02	**0.81 ± 0.01**	0.81 ± 0.01
mBERT-Base	0.90 ± 0.01	0.83 ± 0.02	0.78 ± 0.02	0.80 ± 0.01
# BERT as features				
ptBERT-Large (lin)	0.85	0.70	0.63	0.66
ptBERT-Base (lin)	0.86	0.72	0.68	0.70
mBERT-Base (rbf/poly)	0.86	0.73	0.70	0.71
# BERT as features + LEX				
ptBERT-Large (lin)	0.73	0.24	0.33	0.28
ptBERT-Large (voting)	0.71	0.44	0.45	0.41
ptBERT-Base (rf)	0.67	0.38	0.41	0.39
ptBERT-Base (voting)	0.74	0.41	0.35	0.31
mBERT-Base (poly)	0.75	0.39	0.49	0.43
mBERT-Base (voting)	0.75	0.53	0.52	0.50
# LEX				
lin	0.86	0.75	0.68	0.71

When combining BERT embeddings with lexical features, none of the models were competitive, particularly in F1. Also, on each BERT model, the best accuracy and F1 were achieved by different learning algorithms, except with mBERT-Base embeddings, where the voting model achieved the best results on all metrics. However, we also present the result of mBERT-Base with the polynomial SVM, since it achieves similar results to voting, but implies less computational cost.

Regarding the per class results shown in Table 5.6, performance is competitive in all classes, and proportional to the amount of examples, as in the results for the ASSIN-PTPT corpus. In particular, accuracy per class is identical to the results for ASSIN-PTPT, although unreliable due to class imbalance.

Table 5.6: PI results on ASSIN (PTPT + PTBR), per class and relative to the fine tuned ptBERT-Large model.

Label	Accuracy	Precision	Recall	F1	Examples from Total
neutral	0.94	0.97	0.95	0.96	0.73
entailment	0.93	0.79	0.89	0.84	0.21
paraphrase	0.97	0.85	0.66	0.74	0.06

Considering all corpora, the best performance on PI was achieved in the Portuguese only examples of ASSIN-PTPT, using the fine tuned ptBERT-Large model. However, the per class performances with such model were similar in both the ASSIN-PTPT corpus and the full ASSIN corpus, which have similar distribution of examples per class, although different in total size.

All other systems are based on feature engineering and traditional models, hence equivalent to our traditional models, although none employs BERT. The performance of our traditional models is competitive to all other systems, particularly in accuracy, while our fine tuned models achieve similar or better performance than other systems in most corpora and metrics.

5.2.2 Natural Language Inference

A selection of results for NLI on the ASSIN2 corpus is shown in Table 5.7. When using only BERT embeddings as features, all models achieved similar performance, with any BERT model, except models based on decision trees where performance is not competitive. Using only lexical features, the best performance on all metrics is achieved by voting, but similar results were achieved with random forests and with both non linear SVM models.

For models based on BERT embeddings combined with lexical features, most models achieve similar and competitive performances, except for decision trees on all models, SVM with RBF kernel on ptBERT-Base embeddings, and random forests on ptBERT-Large. With mBERT-Base embeddings, the best result on all metrics was achieved by the voting model, but linear SVM achieves identical or similar results, as shown in Table 5.7, and is considered the best model with such embeddings, since the voting model is computationally more costly. Overall, the best result is achieved with ptBERT-Large embeddings in a model based on the linear SVM, and this result is not similar to any other model based on ptBERT-Large, nor to models based on any other BERT embeddings.

The per class results of the fine tuned ptBERT-Large model are shown in Table 5.8, and indicate that examples classified as not entailment are more often correct than those classified as entailment (from precision), and that overall more entailment examples are correctly predicted than non entailment examples (from recall). The amount of examples per class is not shown, since the corpus is balanced, neither the accuracy, which is equal in both classes and corresponds to the global accuracy previously reported in Table 5.7, since NLI in ASSIN2 is a binary classification task.

Table 5.7: NLI results on ASSIN2.

System	Accuracy	Precision	Recall	F1
# Other systems				
[Santos et al., 2019]	0.67			0.66
[Cabezudo et al., 2019]	0.87			0.87
[Fonseca and Alvarenga, 2019]	0.87			0.87
[Rodrigues et al., 2019a]	0.88			0.88
[Rodrigues et al., 2019b]	0.88			0.88
# BERT fine tuned				
ptBERT-Large	0.89 ± 0.00	**0.87 ± 0.02**	0.93 ± 0.02	**0.90 ± 0.00**
ptBERT-Base	**0.90 ± 0.01**	**0.87 ± 0.02**	**0.94 ± 0.02**	**0.90 ± 0.00**
mBERT-Base	0.88 ± 0.01	0.86 ± 0.01	0.90 ± 0.01	0.88 ± 0.01
# BERT as features				
ptBERT-Large (lin)	0.82	0.79	0.87	0.83
ptBERT-Base (rbf)	0.81	0.75	0.92	0.83
mBERT-Base (rbf)	0.79	0.73	0.90	0.81
# BERT as features + LEX				
ptBERT-Large (lin)	0.83	0.78	0.90	0.84
ptBERT-Base (poly)	0.78	0.72	0.93	0.81
mBERT-Base (lin)	0.78	0.73	0.90	0.81
mBERT-Base (voting)	0.79	0.73	0.91	0.81
# LEX				
rf	0.75	0.73	0.78	0.75
voting	0.76	0.75	0.78	0.77

Table 5.8: NLI results on ASSIN2, per class and relative to the fine tuned ptBERT-Large model.

Label	Precision	Recall	F1
NOT entailment	0.93	0.86	0.90
entailment	0.87	0.94	0.90

A selection of results for the SICK-BR corpus is shown in Table 5.9, and described in the following. SICK-BR is the only corpus where the fine tuned ptBERT-Large model achieves lower performance than all other models, in all metrics, except for models based on lexical features only. However, the best overall result is achieved with a fine tuned model, based on ptBERT-Base. To the best of our knowledge, no other system report results with SICK-BR.

Using only BERT embeddings as features resulted in approximate performances for all models, except for models based on decision trees and random forests, which were not competitive with the remaining models. For mBERT-Base, the best performance is achieved with the SVM based on the RBF kernel, which is similar to the performance achieved with the voting model and the remaining SVM models, but not identical in any metric. With ptBERT-Large embeddings, models based on the linear and polynomial SVM achieve identical performance on most metrics, although the linear SVM is computationally less expensive. With ptBERT-Base, models based on SVM of any kernel achieve identical results, and similar to the best overall result, achieved with the voting model, which is computationally more expensive than any SVM, since it encompasses all models. As such, when using only BERT embeddings as features, and for any ptBERT embeddings, linear SVM is the best choice to build a model, since it achieves results among the best, similar for any ptBERT model, and with the lowest computational effort of all models that achieve similar performance.

Table 5.9: NLI results on SICK-BR.

System	Accuracy	Precision	Recall	F1
# BERT fine tuned				
ptBERT-Large	0.80 ± 0.13	0.72 ± 0.30	0.76 ± 0.24	0.74 ± 0.28
ptBERT-Base	**0.86 ± 0.01**	**0.86 ± 0.01**	**0.85 ± 0.01**	**0.85 ± 0.00**
mBERT-Base	0.84 ± 0.01	0.83 ± 0.01	0.84 ± 0.00	0.84 ± 0.01
# BERT as features				
ptBERT-Large (lin)	0.82	0.83	0.80	0.82
ptBERT-Large (poly)	0.82	0.83	0.81	0.82
ptBERT-Base (voting)	0.82	0.83	0.79	0.81
ptBERT-Base (lin / poly / rbf)	0.81	0.82	0.79	0.81
mBERT-Base (rbf)	0.81	0.82	0.78	0.80
# BERT as features + LEX				
ptBERT-Large (lin)	0.82	0.83	0.80	0.82
ptBERT-Base (lin)	0.81	0.82	0.79	0.80
mBERT-Base (poly)	0.81	0.82	0.79	0.80
# LEX				
rbf	0.78	0.79	0.73	0.75

When combining BERT embeddings with lexical features, all models achieved similar performance, except models based on decision trees and random forests, with any BERT model, and the SVM model based on the RBF kernel with ptBERT-Large embeddings, which were not competitive with the remaining models. Overall, results were similar to those achieved with BERT embeddings only, hence in SICK-BR lexical features had no impact on the performance achieved by BERT alone.

With lexical features only, all models produced similar results, except the model based on decision trees which has not achieved competitive performance. As in the ASSIN2 corpora, and unlike in the PI task, the lowest performance of all models is obtained with lexical features only.

The per class results of the fine tuned ptBERT-Large model on the SICK-BR corpus are shown in Table 5.10, and discussed in the following. Although the mean scores of five instances of this model indicate that it is not the best model in this corpus, as reported in Table 5.9, the per class results are instead computed from an ensemble model computed from the five instances, as previously mentioned in Section 4.2, and results may differ. Also, using the fine tuned ptBERT-Large model in the per class evaluation allows to compare with the results previously reported for the English SICK corpus, since SICK-BR is a translation of SICK, and the fine tuned English BERT-Large model was the best model in SICK, which is most similar in Portuguese to the ptBERT-Large model.

Table 5.10: NLI results on SICK-BR, per class and relative to the fine tuned ptBERT-Large model.

Label	Accuracy	Precision	Recall	F1	Examples from Total
neutral	0.87	0.92	0.85	0.89	0.57
entailment	0.90	0.78	0.91	0.84	0.29
contradiction	0.97	0.90	0.86	0.88	0.14

According to the F1 score, the performance is worse than achieved with the English SICK corpus by at most 0.03 in any class, but follows the same distribution/ranking per class. As such, the analysis previously reported for the per class results in SICK is valid for SICK-BR, and we consider that the ptBERT-Large model achieves equivalent performance to the English BERT-Large model.

5.2.3 Semantic Textual Similarity

A selection of results for STS on the ASSIN-PTPT corpus is shown in Table 5.11, and described in the following. The performance of models based only in lexical features is competitive with that of models based only in BERT embeddings, despite its lower complexity. The best overall performance is obtained with a combination of mBERT-Base embeddings and lexical features.

Table 5.11: STS results on ASSIN-PTPT.

System	MSE	Pearson	Spearman
# Other systems			
[de Souza et al., 2019]	0.64	0.66	
[Freire et al., 2016]	0.72	0.64	
[Hartmann, 2016]	0.66	0.70	
[Alves et al., 2018]	0.43	0.78	
[Santos et al., 2019]	0.63	0.72	
[Pinheiro et al., 2017]	0.57	0.70	
[Barbosa et al., 2016]	0.72	0.64	
[Oliveira Alves et al., 2016]	0.70	0.68	
# BERT fine tuned			
ptBERT-Large	**0.40 ± 0.01**	**0.85 ± 0.01**	**0.83 ± 0.01**
ptBERT-Base	0.47 ± 0.10	**0.85 ± 0.00**	**0.83 ± 0.00**
mBERT-Base	0.53 ± 0.04	0.83 ± 0.01	0.81 ± 0.01
# BERT as features			
ptBERT-Large (poly)	0.55	0.77	0.77
ptBERT-Base (lin)	0.56	0.76	0.76
mBERT-Base (lin)	0.54	0.78	0.77
# BERT as features + LEX			
ptBERT-Large (voting)	0.68	0.79	0.78
ptBERT-Large (rf)	0.65	0.74	0.72
ptBERT-Base (voting)	0.53	0.78	0.77
ptBERT-Base (lin)	0.50	0.76	0.76
mBERT-Base (lin)	0.43	0.79	0.78
# LEX			
rbf	0.57	0.75	0.74

A MSE score above 1 is achieved when combining the ptBERT-Large embeddings with lexical features, which does not occur on other models for such feature set. Overall, the only other occurrence of such MSE score is when using decision trees in models based only on BERT as features. The best MSE score for the combination of ptBERT-Large embeddings and lexical features was obtained with a model based on random forests, which is the only model that considers feature selection. However, for other models combining BERT embeddings and lexical features, the performance of random forests is lower than, for instance, linear SVM, which suggests that the MSE instability with ptBERT-Large embeddings is due to the greater complexity inherent to their greater size, which is reduced when using feature selection, hence the better result achieved with random forests.

For all models employing BERT as features, the performance according to the MSE score is better when using mBERT than with ptBERT embeddings, although we expected the Portuguese-only model to be superior to the multilingual model. When combining any ptBERT embeddings and lexical features, the voting model achieves the best Pearson and Spearman scores, but for ptBERT-Base the results are similar

to those obtained with linear SVM, which is less computationally demanding.

A selection of results for the ASSIN-PTBR corpus is shown in Table 5.12, and described in the following. When using BERT as features, the performance achieved by most models is competitive with the performance from the fine tuned mBERT-Base model. For instance, the best performance with traditional models was obtained when combining BERT embeddings and lexical features, and is better than that of the fine tuned mBERT-Base model. Such result was obtained with the voting model, although similar results were obtained with single models using the same feature set and BERT model, which are less computationally demanding. Using only lexical features resulted in similar performance with all models, and the lowest overall performance.

Table 5.12: STS results on ASSIN-PTBR.

System	MSE	Pearson	Spearman
# Other systems			
[de Souza et al., 2019]	0.45	0.64	
[Santos et al., 2019]	0.37	0.71	
[Freire et al., 2016]	0.47	0.62	
[Silva and Rigo, 2018]	0.43	0.66	
[Hartmann, 2016]	0.38	0.70	
[Alves et al., 2018]	0.34	0.74	
[Barbosa et al., 2016]	0.44	0.65	
[Oliveira Alves et al., 2016]	0.44	0.65	
[Pinheiro et al., 2017]	0.37	0.71	
# BERT fine tuned			
ptBERT-Large	**0.23 ± 0.01**	**0.84 ± 0.01**	**0.83 ± 0.01**
ptBERT-Base	0.25 ± 0.01	0.83 ± 0.00	0.82 ± 0.00
mBERT-Base	0.32 ± 0.02	0.78 ± 0.01	0.77 ± 0.01
# BERT as features			
ptBERT-Large (lin / poly / rbf)	0.30	0.78	0.77
ptBERT-Base (lin)	0.29	0.79	0.79
mBERT-Base (lin / poly / rbf)	0.31	0.77	0.76
# BERT as features + LEX			
ptBERT-Large (rbf)	0.35	0.78	0.78
ptBERT-Large (voting)	0.34	0.80	0.79
ptBERT-Base (lin)	0.29	0.79	0.79
ptBERT-Base (voting)	0.28	0.80	0.79
mBERT-Base (poly / voting)	0.30	0.78	0.77
# LEX			
poly / rbf	0.34	0.74	0.73

All models based only on BERT as features performed similarly in all metrics and with any BERT model, except for models based on decision trees and random forests which achieved lower performance than the remaining. With ptBERT-Large and mBERT-Base embeddings only, the results were identical in models based on SVM with any kernel, hence no advantage is achieved from non linear kernels, which are more computationally demanding, and the linear SVM is considered the best model in such feature set. When combining BERT embeddings with lexical features, performance was similar in all models, except for models based on decision trees with any BERT model, and SVM based on linear and polynomial kernels with ptBERT-Large embeddings, where performance is not competitive, particularly in MSE.

A selection of results for the full ASSIN corpus is shown in Table 5.13, and described in the following.

Unlike in most other corpora, the best MSE was achieved with only BERT as features, and not by a fine tuned model. This is also the only corpus where all models based on the combination of BERT embeddings and lexical features achieve lower performance in all metrics than any model based only on either of these feature sets. Performance in models based only in lexical features is competitive with the performance from models based only on BERT embeddings, although inferior.

Table 5.13: STS results on ASSIN (PTPT + PTBR).

System	MSE	Pearson	Spearman
# Other systems			
[de Souza et al., 2019]	0.56	0.66	
[Freire et al., 2016]	0.59	0.62	
[Hartmann, 2016]	0.52	0.68	
[Barbosa et al., 2016]	0.59	0.63	
[Oliveira Alves et al., 2016]	0.57	0.65	
[Pinheiro et al., 2017]	0.47	0.70	
# BERT fine tuned			
ptBERT-Large	0.48 ± 0.11	**0.82 ± 0.01**	**0.81 ± 0.01**
ptBERT-Base	0.49 ± 0.03	0.81 ± 0.01	0.80 ± 0.01
mBERT-Base	0.43 ± 0.04	0.80 ± 0.01	0.79 ± 0.01
# BERT as features			
ptBERT-Large (poly)	0.42	0.76	0.76
ptBERT-Base (poly)	**0.41**	0.77	0.76
mBERT-Base (poly / rbf)	**0.41**	0.77	0.76
# BERT as features + LEX			
ptBERT-Large (voting)	0.90	0.55	0.57
ptBERT-Base (voting)	0.67	0.64	0.66
mBERT-Base (voting)	0.58	0.70	0.70
# LEX			
rbf	0.45	0.73	0.71

Using only BERT embeddings as features resulted in competitive performance from all models, except for models based on decision trees and random forests which achieved lower performance than the remaining. Using feature vectors of BERT embeddings combined with lexical features, all models failed, particularly in the MSE metric, where the most competitive score is achieved by a voting model based on mBERT-Base embeddings. For instance, with SVM based models, most MSE scores were above 1. All models based only on lexical features achieved similar performance, except models based on decision trees which were not competitive with the remaining.

A selection of results for the ASSIN2 corpus is shown in Table 5.14, and described in the following. Regarding models based only in lexical features, performance is similar on all models except decision trees and SVM with RBF kernel, and is worst than the performance achieved with models involving BERT embeddings. In particular, the best overall result from traditional models is achieved with BERT embeddings only.

Using only BERT as features, the models for each BERT model achieve similar performance, except models based on decision trees, which are not competitive. The best performances are achieved with models based on non linear SVM, for ptBERT-Large, and voting for the remaining BERT models, although such results are similar or identical to the performance achieved with linear SVM on all BERT models, which is less computationally demanding.

Table 5.14: STS results on ASSIN2.

System	MSE	Pearson	Spearman
# Other systems			
[de Souza et al., 2019]	0.60	0.72	
[Santos et al., 2019]	0.58	0.73	
[Cabezudo et al., 2019]	0.64	0.73	
[Fonseca and Alvarenga, 2019]	**0.39**	0.80	
[Rodrigues et al., 2019a]	0.52	0.83	
[Rodrigues et al., 2019b]	0.59	0.79	
# BERT fine tuned			
ptBERT-Large	0.50 ± 0.09	**0.84 ± 0.01**	**0.81 ± 0.01**
ptBERT-Base	0.43 ± 0.03	**0.84 ± 0.01**	0.80 ± 0.01
mBERT-Base	0.49 ± 0.04	0.82 ± 0.01	0.79 ± 0.00
# BERT as features			
ptBERT-Large (lin)	0.48	0.79	0.73
ptBERT-Large (poly / rbf)	0.49	0.81	0.75
ptBERT-Base (lin)	0.50	0.79	0.74
ptBERT-Base (voting)	0.50	0.81	0.76
mBERT-Base (lin)	0.56	0.75	0.69
mBERT-Base (voting)	0.56	0.76	0.72
# BERT as features + LEX			
ptBERT-Large (rf)	0.54	0.71	0.66
ptBERT-Large (voting)	2.33	0.78	0.70
ptBERT-Base (rf)	0.49	0.75	0.72
ptBERT-Base (voting)	0.91	0.79	0.72
mBERT-Base (rf)	0.51	0.74	0.70
mBERT-Base (voting)	0.43	0.77	0.72
# LEX			
rf	0.60	0.70	0.68

When combining ptBERT-Base or ptBERT-Large embeddings with lexical features, only the model based on random forests achieved competitive performance in the MSE metric, while for the remaining metrics results are similar in most models, and the best performance is achieved with the voting model. With mBERT-Base embeddings, the best result in all metrics is achieved with the voting model, although similar to the result with random forests, which is less computationally demanding since it only involves computing one model.

A selection of results for the SICK-BR corpus is shown in Table 5.15, and described in the following. As in the results for NLI on this corpus, and unlike in both NLI and STS on the equivalent SICK corpus, the fine tuned model based on BERT-Large embeddings (here, ptBERT-Large) achieves one of the lowest performance results. However, the best results in traditional models are achieved with only BERT as features, and these are competitive with the best result achieved by fine tuned models.

Using only BERT as features resulted in similar performance for all models, except for models based on decision trees and random forests in all BERT models, and linear SVM with mBERT-Base embeddings, which were not competitive when compared to the remaining models of each BERT model. With only lexical features, all models achieve similar performance, except decision trees and linear SVM, which achieve lower performance scores.

Table 5.15: STS results on SICK-BR.

System	MSE	Pearson	Spearman
# BERT fine tuned			
ptBERT-Large	0.47 ± 0.42	0.70 ± 0.39	0.66 ± 0.37
ptBERT-Base	0.30 ± 0.02	**0.86 ± 0.01**	**0.80 ± 0.01**
mBERT-Base	0.37 ± 0.02	0.85 ± 0.00	0.79 ± 0.00
# BERT as features			
ptBERT-Large (poly / rbf)	**0.28**	0.85	0.79
ptBERT-Base (rbf)	0.30	0.84	0.77
mBERT-Base (rbf)	0.33	0.82	0.75
# BERT as features + LEX			
ptBERT-Large (rf)	0.44	0.77	0.70
ptBERT-Large (voting)	0.54	0.85	0.78
ptBERT-Base (rf)	0.40	0.79	0.72
ptBERT-Base (voting)	0.37	0.82	0.74
mBERT-Base (rbf)	0.36	0.81	0.73
mBERT-Base (voting)	0.34	0.82	0.76
# LEX			
rf	0.45	0.75	0.68
voting	0.45	0.75	0.69

For the combination of BERT embeddings and lexical features, models based on decision trees were not competitive, in all metrics and BERT models. With ptBERT-Large embeddings combined with lexical features, results with SVM models based on the RBF kernel were not competitive in all metrics, and with linear and polynomial kernels results were not competitive in the MSE metric, which achieved scores greater than 1, although the remaining metrics for such models achieve competitive results. When using lexical features, combined or not, the voting model achieves the best results on some metrics of each different feature set. However, for the performance of each voting model there is a single model with similar performance, and less computationally demanding.

5.3 Discussion

In fine tuned models, we measured statistical significance between the values returned by each pair of models, using a t-test (five runs for each model) for all metrics. Considering $p = 0.05$, there are statistically significant differences between models ptBERT-Base and mBERT-Base, and also between models ptBERT-Large and mBERT-Base, for most evaluation metrics and tasks. Exceptions include, for instance, the accuracy metric at PI in the ASSIN-PTPT corpus, for the former pair of models, and most evaluation metrics in both NLI and STS with the SICK-BR corpus, for the latter pair of models. Regarding the differences between models ptBERT-Base and ptBERT-Large, results varied, since we found statistically significant differences for some metrics, but not for others, on all tasks.

Fine tuned models achieved better results than traditional models in most of the evaluated corpora and tasks. Particularly, the ptBERT-Large fine tuned model achieved the best results in most corpora and tasks, compared to our other setups and to other systems. However, for instance, at NLI in SICK-BR it achieved the worst overall performance in precision and F1, and for STS in ASSIN (PTPT + PTBR) it failed to achieve the best MSE, being surpassed by traditional models based on any BERT embeddings. In the following we further investigate this model.

Regarding the STS task, the results of the ptBERT-Large fine tuned model for Pearson and Spearman correlation coefficients are greater than 0.8 in most evaluated corpora, which reveals a strong correlation between predictions and true values, indicating that the predictions for most examples are distant from their true values in approximately the same magnitude.

To further study the performance of the ptBERT-Large fine tuned model, we employed the ensemble of five instances of this model, wherein the predictions from each instance were averaged, and the MSE was computed on said average. The ptBERT-Large fine tuned model mentioned in the following corresponds to said ensemble.

Interestingly, in examples with greater distance between predictions and true values, this difference is almost constant, as shown in Figure 5.1 for the ASSIN2 corpus, and the predictions are greater than the true values, suggesting that the STS predictions of ptBERT-Large fine tuned model are overly confident. As such, we did an experiment in the ASSIN2 corpus, wherein the MSE score of the ptBERT-Large fine tuned model is worse than in some of our traditional models also based on BERT embeddings, by subtracting a constant value whenever the difference between prediction and true was greater than a certain value. The best results were obtained by subtracting 0.9 from the prediction, when the difference between the prediction and the true value was greater than 0.5. The resulting MSE was 0.11, as opposed to the 0.47 originally obtained with the ensemble model, and the prediction was subtracted in 1014 of the 2448 test examples. As this condition implies knowing the true value, we also experimented with subtracting 0.9 on all predictions, which resulted in a MSE of 0.43.

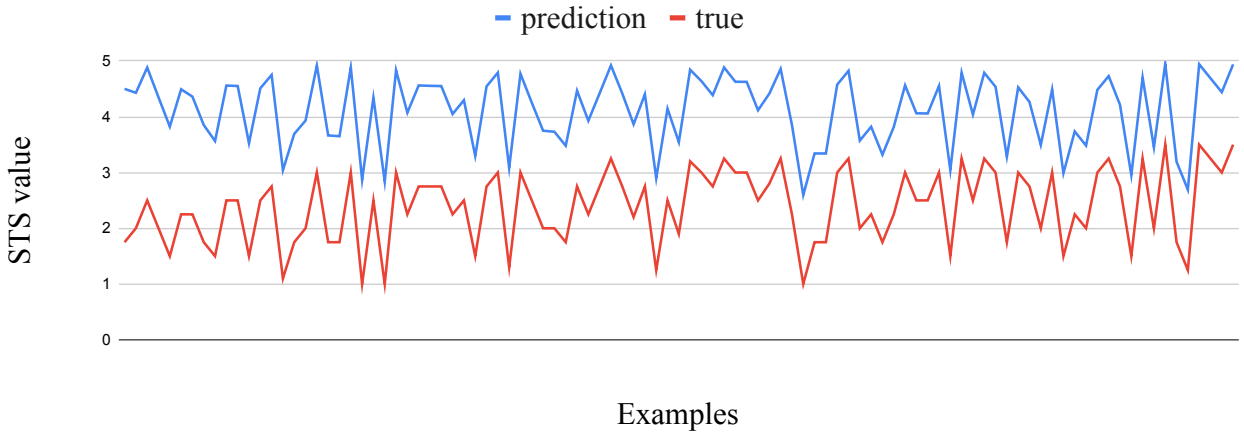


Figure 5.1: Top 100 examples of ASSIN2 with greater distance between predicted and true values of the STS task, where such distance is greater than 0.5.

We also performed the same analysis for the ASSIN-PTBR corpus, wherein the ptBERT-Large fine tuned model achieved the best MSE value of all corpora. Here, the distance between predictions and true values is also approximately constant, as shown in Figure 5.2. Again, we experimented with the previously mentioned conditional subtraction, but here the best MSE was obtained by subtracting 0.8 from the predicted value whenever the distance between original prediction and true value was greater than 0.5. The resulting MSE was 0.12, instead of the original 0.21 of the ensemble model, and the condition complied with 285 of the 2000 test examples. Subtracting 0.8 from all predictions resulted in a MSE of 0.8, which is worse than the original result, since only approximately 10% of the test examples complied with the condition for subtraction.

Moreover, we inspected individual examples where the ptBERT-Large fine tuned model failed to identify the NLI class. We did not find particular differences between the language employed in such cases and

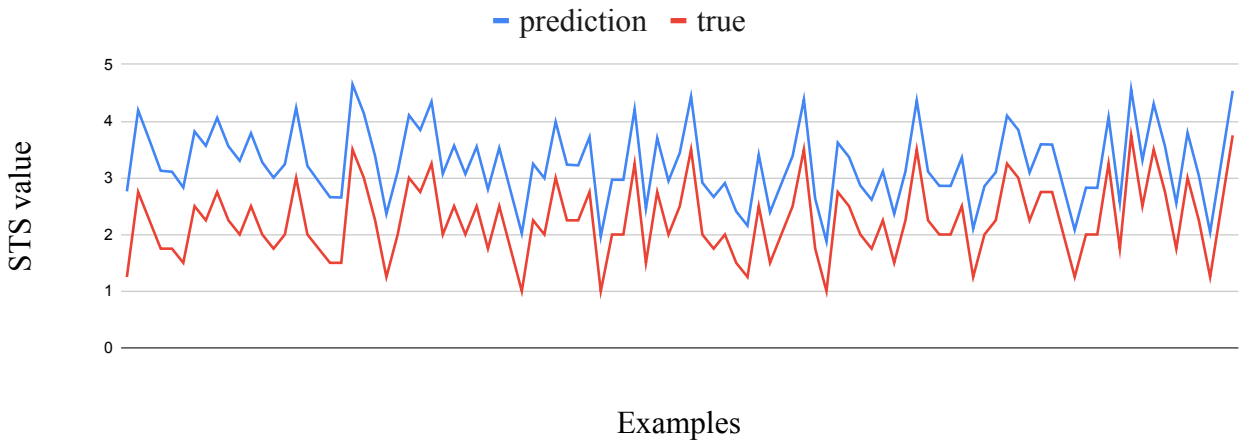


Figure 5.2: Top 100 examples of ASSIN-PTBR with greater distance between the prediction and true values of the STS task, where such distance is greater than 0.5.

that of successful classification cases. As embeddings are not interpretable, and hence do not provide an explanation from their features, we were not able to reason about the language in misclassifications. However, it was possible to observe that some examples from the corpora are difficult to understand. For instance, in ASSIN2, sentences *Um peixe está sendo cortado por um cara* and *Um cara está fatiando um peixe* are not considered as entailment, but *A comida nas bandejas está sendo comida pelos filhotes de gato* and *Poucos filhotes de gato estão comendo* are considered as entailment. However, it is out of the scope of this paper to discuss the quality of the corpora, although issues with NLI corpora can be found in [Kalouli et al., 2019], particularly regarding the development guidelines of the SICK corpus [Marelli et al., 2014b], on which our evaluated ASSIN2 and SICK-BR corpora were based.

6

Conclusion

In this thesis we targeted the problem of assessing if two sentences are semantically related, such as to determine if they are equivalent. We designed and presented various models to address such problem for English and Portuguese, based on lexical and semantic features. The performance of our models was evaluated in the tasks of PI, NLI and STS, using various corpora for each language, and the obtained results are competitive with other approaches.

We introduced a new set of semantic features for English, computed from DRS, and explored a set of lexical features for English and Portuguese, previously compiled by other authors [Marques, 2015]. We also leveraged BERT embeddings as semantic features, and for Portuguese this was the only form of semantics, since DRS are not available for Portuguese, to the best of our knowledge. Our models are based on combinations of embeddings, lexical and semantic features, and implement various machine learning techniques, such as optimal parameter search, model combination by voting, and fine tuning of deep learning models.

Most current approaches for our target tasks are based on large-scale models, such as BERT, which are computationally demanding, while our models based on lexical and semantic features are suitable for computation in most computers. Nonetheless, we evaluate large-scale models, based on fine tuning BERT,

and while these achieve the best overall results, in some tasks and languages the results achieved with our features in traditional models are competitive with those achieved with such large-scale models.

6.1 Research Questions Review

Our research questions, previously introduced in Section 1.2, are reviewed in the following:

- RQ1: In what extent can lexical and semantic features contribute to the tasks of PI, NLI and STS, both isolated and combined?

  For all tasks in English, models combining BERT embeddings with both lexical and semantic features achieve better performance than models using only generic embeddings. Moreover, except for the STS task, models based only on the combination of lexical and semantic features achieve better performance than models based only in embeddings, although the latter are more complex and computationally demanding.

  The performance of models based only on either lexical or semantic features improves when combined with BERT embeddings, for all tasks in English. Also, the performance of models based on generic embeddings and/or lexical features improves when combined with semantic features, such as the performance of models based on embeddings and/or semantic features improves when combined with lexical features. The only notable performance gain of combining lexical or semantic features with embeddings, relative to using only embeddings, is achieved with the combination of embeddings and lexical features in the PI task. In all other tasks, combining embeddings with either lexical or semantic features results in similar performance.

  Notably, for the PI and NLI classification tasks in English corpora, models based on the combination of lexical and semantic features achieve competitive performance, even without considering BERT embeddings. In particular, for the PI task, the performance of models based only on lexical features is competitive with that of the best models based on generic features, and represents the best performance for the least computational effort. For the STS regression task, embeddings are required for models to achieve competitive performance.

  For Portuguese, PI is the only task where the performance of models based only in lexical features is better than any other model, except for fine tuned models. In other tasks, models based only in generic BERT embeddings achieve the best performance for the least computational effort, since combining embeddings with lexical features does not result in notable performance gains. In some corpora of the STS task, such combination even results in worst performance than using only embeddings or lexical features.

  As such, lexical and semantic features contribute to the tasks of PI, NLI and STS, both isolated and combined, by providing competitive performance with less computational effort than BERT embeddings, although such performance is not consistent for all tasks, corpora or languages.

- RQ2: How to extract semantic features from DRS, which can contribute to the tasks of PI, NLI and STS for the English language, and how can these be combined with other types of feature?

  We design our semantic features from DRS mostly by observing the graphical representation of DRS, in the boxed format previously described in Section 3.2. From the inspection of two DRS, one for each sentence of an example pair, we identify common aspects, for instance in logic predicates and the structure of boxes. Most features are based on counting common aspects and calculating percentages of such counts, both for the whole DRS and for particular inner structures, such as the groups of predicates labelled as part of negations and implications. As such, our semantic features

model the semantic equivalence between sentences, as derived from the semantics in DRS, which all of our target tasks address in some form.

Our semantic features comprise different types of numbers, namely discrete values from counting and continuous values from percentages. We scale each value according to the type of number, such that a combination with other features is possible by appending our semantic features to a given feature vector.

- RQ3: In our target tasks, what is the performance of pre-trained models for languages other than English, in particular for the Portuguese language, and how is the performance affected when combining lexical features with such models?

  Our models based only on generic pre-trained BERT models, both from multilingual or Portuguese variants, achieve better results than models based only in lexical features, for all tasks on Portuguese data, except for the PI task where performance is similar or better when using lexical features only. Overall, the best performance is achieved with the fine tuned model, which is based on the Portuguese variant.

  Using the multilingual or Portuguese variants results in similar performance, although the best results are most often achieved with the Portuguese model. In particular, for one of the evaluation metrics in the STS task, the generic Portuguese model achieves better results than the fine tuned version of the same model, on various corpora.

  The performance achieved by using a combination of pre-trained models and lexical features is similar to that of using only pre-trained models, in most tasks and corpora. However, for some corpora on each task, the best performance is obtained with such combination. For instance, for one of the metrics of the STS task, models based on a combination of pre-trained models and lexical features achieve better performance than using only pre-trained models or lexical features, and identical results to those of fine tuned models. As such, lexical features contribute to improve the performance of pre-trained models, at least on some corpora.

6.2 Contributions Review

With this thesis we contribute a new set of semantic features to address tasks that rely on assessing some form of equivalence or semantic relation between two English sentences. Our features are computed from the logic-based description of semantics provided in DRS, and leverage symbolic, structural and natural language components in such formal representation. For instance, some of our features are based on the equivalency between tokens in a pair of logic predicates, where some consider symbols of the DRS formalism, and others only consider words from the sentence. Different forms of equivalency are considered, some of which combine lexical resources, such as embeddings and WordNet, with logic predicates, for instance to achieve a form of soft matching, similarly to the logic unification between vector representations introduced in other works [Rocktäschel and Riedel, 2017]. Structure-wise, some of our features model, for instance, the common aspects between negations and implications of each sentence, which are represented in a DRS as inner groups of predicates.

We present an evaluation of our semantic features on the tasks of PI, NLI and STS, using corpora with distinct types of language and domains, and also combining such semantic features with lexical features and embeddings. Results show that the performance of our semantic features is competitive with the performance obtained with lexical features or embeddings, but not superior to any of these. However, the best performance on some evaluation metrics and corpora were achieved with a combination of our semantic features and some of the remaining types of feature considered.

Another contribution of this thesis is an assessment on the performance of models that address some form of sentence similarity in Portuguese, relying on embeddings and lexical features, and evaluated on the tasks of PI, NLI and STS. In particular, we evaluate the BERT model, known for state of the art performance with English corpora, and conclude that its multilingual version achieves competitive results on all of our target tasks, although not superior to the performance obtained with a BERT model pre-trained only with Portuguese data.

Furthermore, we reviewed and updated the lexical features for English and Portuguese, previously introduced in [Marques, 2015], and assessed their performance on more tasks, and the latest Portuguese corpora. In particular, for Portuguese, we conclude that the performance obtained with PI models based only in lexical features is better than the performance from most models based on generic embeddings. For NLI and STS, the performance of models based only in lexical features is not competitive with the remaining models, but combining lexical features with generic embeddings achieve better performance than models based only in generic embeddings. Similar conclusions were drawn for English.

6.3 Future Work

Semantic features is where most of our future work is targeted, particularly in logic-based forms of language analysis and their application in models. For instance, some works combine logic, probabilities and neural models [Manhaeve et al., 2018] [Winters et al., 2022]. Also, more modern forms of generating DRS could be explored, for instance based on neural networks [van Noord et al., 2018b, Abzianidze et al., 2019], to evaluate their performance on computing our features for English, and enable usage of DRS semantics in Portuguese [Liu et al., 2021]. Moreover, a combination of generic BERT models and DRS [van Noord et al., 2020] could be envisaged.

Future work for Portuguese includes exploring other resources available to compute features, such as from syntactic parsing [Silva et al., 2010], Portuguese versions of WordNet [de Paiva et al., 2012, Branco et al., 2020], dependency parsing [Mamede et al., 2012] and semantic role labeling [Sequeira et al., 2012, Talhadas, 2013, Fonseca and Rosa, 2013]. For English, further exploration on how to leverage syntactic features could be envisaged, for instance as computed with tree kernels, which we introduced in a previous work [Fialho et al., 2019].

Regarding model manipulation, we envisage performance improvements from further searching of optimal parameters to train our models, and exploring other types of model to employ our generic features, such as models based on neural networks. In particular, regularization techniques for neural networks could be explored, based on the logic predicates in DRS [Minervini and Riedel, 2018].

At last, further analysis of language in examples may explain the success or failure of our models in particular formulations of language. Moreover, an evaluation of our features in corpora with other types of language could be explored, for instance with frequently asked questions in Portuguese [Gonçalo Oliveira et al., 2020, Carriço and Quaresma, 2021]. Also, some form of adaptation for corpora designed for other tasks could be explored, such as designing a PI task from the question/interaction alternatives in the knowledge base of conversational agents [Ventura et al., 2020, Rodrigues et al., 2022].

Bibliography

[Abzianidze, 2015] Abzianidze, L. (2015). A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2492–2502, Lisbon, Portugal. Association for Computational Linguistics.

[Abzianidze et al., 2017] Abzianidze, L., Bjerva, J., Evang, K., Haagsma, H., van Noord, R., Ludmann, P., Nguyen, D.-D., and Bos, J. (2017). The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

[Abzianidze et al., 2019] Abzianidze, L., van Noord, R., Haagsma, H., and Bos, J. (2019). The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

[Agirre et al., 2015] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., Rigau, G., Uria, L., and Wiebe, J. (2015). SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

[Agirre et al., 2014] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., and Wiebe, J. (2014). SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

[Agirre et al., 2016] Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., and Wiebe, J. (2016). SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

[Agirre et al., 2012] Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. (2012). SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

[Ahmed and Mercer, 2020] Ahmed, M. and Mercer, R. E. (2020). Modelling sentence pairs via reinforcement learning: An actor-critic approach to learn the irrelevant words. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7358–7366.

[Alves et al., 2018] Alves, A., Oliveira, H. G., Rodrigues, R., and Encarnação, R. (2018). ASAPP 2.0: Advancing the state-of-the-art of semantic textual similarity for Portuguese. In Henriques, P. R., Leal, J. P., Leitão, A. M., and Guinovart, X. G., editors, *7th Symposium on Languages, Applications and Technologies (SLATE 2018)*, volume 62 of *OpenAccess Series in Informatics (OASIcs)*, pages 12:1–12:17, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Ameixa et al., 2014] Ameixa, D., Coheur, L., Fialho, P., and Quaresma, P. (2014). Luke, i am your father: Dealing with out-of-domain requests by using movies subtitles. In Bickmore, T., Marsella, S., and Sidner, C., editors, *Intelligent Virtual Agents*, pages 13–21, Cham. Springer International Publishing.

[Anchiêta et al., 2020] Anchiêta, R. T., de Sousa, R. F., and Pardo, T. A. S. (2020). Modeling the paraphrase detection task over a heterogeneous graph network with data augmentation. *Inf.*, 11(9):422.

[Androutsopoulos and Malakasiotis, 2010] Androutsopoulos, I. and Malakasiotis, P. (2010). A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38(1):135–187.

[Arase and Tsujii, 2021] Arase, Y. and Tsujii, J. (2021). Transfer fine-tuning of bert with phrasal paraphrases. *Computer Speech & Language*, 66:101164.

[Banarescu et al., 2013] Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

[Bar-Haim et al., 2006] Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., and Szpektor, I. (2006). The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*.

[Bar-Haim et al., 2014] Bar-Haim, R., Dagan, I., and Szpektor, I. (2014). Benchmarking applied semantic inference: The PASCAL recognising textual entailment challenges. In Dershowitz, N. and Nissan, E., editors, *Language, Culture, Computation. Computing - Theory and Technology - Essays Dedicated to Yaacov Choueka on the Occasion of His 75th Birthday, Part I*, volume 8001 of *Lecture Notes in Computer Science*, pages 409–424. Springer.

[Barbosa et al., 2016] Barbosa, L., Cavalin, P., Guimarães, V., and Kormaksson, M. (2016). Blue man group no assin: Usando representações distribuídas para similaridade semântica e inferência textual. *Linguamática*, 8(2):15–22.

[Basile, 2015] Basile, V. (2015). *From logic to language: Natural language generation from logical forms*. PhD thesis, University of Groningen.

[Beltagy et al., 2014] Beltagy, I., Erk, K., and Mooney, R. (2014). Semantic parsing using distributional semantics and probabilistic logic. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 7–11, Baltimore, MD. Association for Computational Linguistics.

[Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

[Bhagat and Hovy, 2013] Bhagat, R. and Hovy, E. (2013). What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

[Bird and Loper, 2004] Bird, S. and Loper, E. (2004). NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

[Bjerva et al., 2014] Bjerva, J., Bos, J., van der Goot, R., and Nissim, M. (2014). The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 642–646, Dublin, Ireland. Association for Computational Linguistics.

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

[Bond et al., 2020] Bond, F., Morgado da Costa, L., Goodman, M. W., McCrae, J. P., and Lohk, A. (2020). Some issues with building a multilingual Wordnet. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3189–3197, Marseille, France. European Language Resources Association.

[Bos, 2008] Bos, J. (2008). Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 277–286. College Publications.

[Bos, 2015] Bos, J. (2015). Open-domain semantic parsing with boxer. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 301–304, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

[Bos, 2016] Bos, J. (2016). Expressive Power of Abstract Meaning Representations. *Computational Linguistics*, 42(3):527–535.

[Bos and Markert, 2005] Bos, J. and Markert, K. (2005). Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

[Bowman et al., 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

[Branco et al., 2020] Branco, A., Grilo, S., Bolrinha, M., Saedi, C., Branco, R., Silva, J., Querido, A., de Carvalho, R., Gaudio, R., Avelãs, M., and Pinto, C. (2020). The MWN.PT WordNet for Portuguese: Projection, validation, cross-lingual alignment and distribution. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4859–4866, Marseille, France. European Language Resources Association.

[Brown et al., 1992] Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

[Brun and Hagège, 2003] Brun, C. and Hagège, C. (2003). Normalization and paraphrasing using symbolic methods. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 41–48, Sapporo, Japan. Association for Computational Linguistics.

[Cabezudo et al., 2019] Cabezudo, M. A. S., Inácio, M., Rodrigues, A. C., Casanova, E., and de Sousa, R. F. (2019). NILC at ASSIN 2: Exploring multilingual approaches. In Oliveira, H. G., Real, L., and Fonseca, E., editors, *Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, BA, Brazil, October 15, 2019*, volume 2583 of *CEUR Workshop Proceedings*, pages 49–58. CEUR-WS.org.

[Camburu et al., 2018a] Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T., and Blunsom, P. (2018a). e-snli: Natural language inference with natural language explanations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc.

[Camburu et al., 2018b] Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T., and Blunsom, P. (2018b). e-snli: Natural language inference with natural language explanations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Carriço and Quaresma, 2021] Carriço, N. and Quaresma, P. (2021). Sentence Embeddings and Sentence Similarity for Portuguese FAQs. In *Proc. IberSPEECH 2021*, pages 200–204.

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

[Chandrasekaran and Mago, 2021] Chandrasekaran, D. and Mago, V. (2021). Evolution of semantic similarity—a survey. *ACM Comput. Surv.*, 54(2).

[Chawla, 2005] Chawla, N. V. (2005). Data mining for imbalanced datasets: An overview. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, pages 853–867. Springer US, Boston, MA.

[Chen and Cherry, 2014] Chen, B. and Cherry, C. (2014). A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

[Chen et al., 2021a] Chen, Q., Ji, F., Zeng, X., Li, F.-L., Zhang, J., Chen, H., and Zhang, Y. (2021a). KACE: Generating knowledge aware contrastive explanations for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2516–2527, Online. Association for Computational Linguistics.

[Chen et al., 2017] Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. (2017). Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

[Chen et al., 2021b] Chen, Z., Gao, Q., and Moss, L. S. (2021b). NeuralLog: Natural language inference with joint neural and logical reasoning. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 78–88, Online. Association for Computational Linguistics.

[Cheng and Kartsaklis, 2015] Cheng, J. and Kartsaklis, D. (2015). Syntax-aware multi-sense word embeddings for deep compositional models of meaning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1531–1542, Lisbon, Portugal. Association for Computational Linguistics.

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

[Clark and Curran, 2004] Clark, S. and Curran, J. R. (2004). Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 103–110, Barcelona, Spain.

[Conneau and Kiela, 2018] Conneau, A. and Kiela, D. (2018). SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

[Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.

[Creutz, 2018] Creutz, M. (2018). Open subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

[Curran et al., 2007] Curran, J., Clark, S., and Bos, J. (2007). Linguistically motivated large-scale NLP with C&C and boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic. Association for Computational Linguistics.

[Dagan et al., 2009] Dagan, I., Dolan, B., Magnini, B., and Roth, D. (2009). Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 15(4):i–xvii.

[Dakota and Kübler, 2016] Dakota, D. and Kübler, S. (2016). From discourse representation structure to event semantics: A simple conversion? In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 343–352.

[Das and Smith, 2009] Das, D. and Smith, N. A. (2009). Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 468–476, Suntec, Singapore. Association for Computational Linguistics.

[de Paiva et al., 2012] de Paiva, V., Rademaker, A., and de Melo, G. (2012). OpenWordNet-PT: An open Brazilian Wordnet for reasoning. In *Proceedings of COLING 2012: Demonstration Papers*, pages 353–360, Mumbai, India. The COLING 2012 Organizing Committee.

[de Souza et al., 2019] de Souza, J. V. A., e Oliveira, L. E. S., Gumiel, Y. B., Carvalho, D. R., and Moro, C. M. C. (2019). Incorporating multiple feature groups to a siamese neural network for semantic textual similarity task in portuguese texts. In Oliveira, H. G., Real, L., and Fonseca, E., editors, *Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, BA, Brazil, October 15, 2019*, volume 2583 of *CEUR Workshop Proceedings*, pages 59–68. CEUR-WS.org.

[Denkowski and Lavie, 2014] Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

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

[Dey et al., 2016] Dey, K., Shrivastava, R., and Kaushik, S. (2016). A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2880–2890, Osaka, Japan. The COLING 2016 Organizing Committee.

[Dolan and Brockett, 2005] Dolan, B. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing.

[Felkin, 2007] Felkin, M. (2007). Comparing classification results between n-ary and binary problems. In Guillet, F. and Hamilton, H. J., editors, *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*, pages 277–301. Springer.

[Fellbaum, 1998] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

[Fernando and Stevenson, 2008] Fernando, S. and Stevenson, M. (2008). A semantic similarity approach to paraphrase detection. In *Proceedings of the Annual Research Colloquium on Computational Linguistics in the UK*.

[Fialho et al., 2013] Fialho, P., Coheur, L., Curto, S., Cláudio, P., Costa, Â., Abad, A., Meinedo, H., and Trancoso, I. (2013). Meet EDGAR, a tutoring agent at MONSERRATE. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–66, Sofia, Bulgaria. Association for Computational Linguistics.

[Fialho et al., 2019] Fialho, P., Coheur, L., and Quaresma, P. (2019). From Lexical to Semantic Features in Paraphrase Identification. In Rodrigues, R., Janousek, J., Ferreira, L., Coheur, L., Batista, F., and Oliveira, H. G., editors, *8th Symposium on Languages, Applications and Technologies (SLATE 2019)*, volume 74 of *OpenAccess Series in Informatics (OASIcs)*, pages 9:1–9:11, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Fialho et al., 2020a] Fialho, P., Coheur, L., and Quaresma, P. (2020a). Back to the feature, in entailment detection and similarity measurement for portuguese. In Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., and Gonçalves, T., editors, *Computational Processing of the Portuguese Language*, pages 164–173, Cham. Springer International Publishing.

[Fialho et al., 2020b] Fialho, P., Coheur, L., and Quaresma, P. (2020b). Benchmarking natural language inference and semantic textual similarity for portuguese. *Information*, 11(10).

[Fialho et al., 2020c] Fialho, P., Coheur, L., and Quaresma, P. (2020c). To bert or not to bert dealing with possible bert failures in an entailment task. In Lesot, M.-J., Vieira, S., Reformat, M. Z., Carvalho, J. P., Wilbik, A., Bouchon-Meunier, B., and Yager, R. R., editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 734–747, Cham. Springer International Publishing.

[Fialho et al., 2016] Fialho, P., Marques, R., Martins, B., Coheur, L., and Quaresma, P. (2016). Inesc-id@assin: Medição de similaridade semântica e reconhecimento de inferência textual. *Linguamática*, 8(2):33–42.

[Fialho et al., 2017] Fialho, P., Patinho Rodrigues, H., Coheur, L., and Quaresma, P. (2017). L2F/INESC-ID at SemEval-2017 tasks 1 and 2: Lexical and semantic features in word and textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 213–219, Vancouver, Canada. Association for Computational Linguistics.

[Filice et al., 2015] Filice, S., Da San Martino, G., and Moschitti, A. (2015). Structural representations for learning relations between pairs of texts. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1003–1013, Beijing, China. Association for Computational Linguistics.

[Filice and Moschitti, 2016] Filice, S. and Moschitti, A. (2016). Learning to recognize ancillary information for automatic paraphrase identification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1109–1114, San Diego, California. Association for Computational Linguistics.

[Finch et al., 2005] Finch, A., Hwang, Y.-S., and Sumita, E. (2005). Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In *Proceedings of the International Workshop on Paraphrasing*.

[Flanigan et al., 2014] Flanigan, J., Thomson, S., Carbonell, J., Dyer, C., and Smith, N. A. (2014). A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

[Fonseca and Alvarenga, 2019] Fonseca, E. and Alvarenga, J. P. R. (2019). Multilingual transformer ensembles for portuguese natural language tasks. In Oliveira, H. G., Real, L., and Fonseca, E., editors, *Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, BA, Brazil, October 15, 2019*, volume 2583 of *CEUR Workshop Proceedings*, pages 68–77. CEUR-WS.org.

[Fonseca et al., 2016] Fonseca, E. R., Borges dos Santos, L., Criscuolo, M., and Aluísio, S. M. (2016). Overview of the evaluation of semantic similarity and textual inference. *Linguamática*, 8(2):3–13.

[Fonseca and Rosa, 2013] Fonseca, E. R. and Rosa, J. L. G. (2013). A two-step convolutional neural network approach for semantic role labeling. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

[Freire et al., 2016] Freire, J., Pinheiro, V., and Feitosa, D. (2016). Flexsts: Um framework para similaridade semântica textual. *Linguamática*, 8(2):23–31.

[Gonçalo Oliveira et al., 2020] Gonçalo Oliveira, H., Ferreira, J., Santos, J., Fialho, P., Rodrigues, R., Coheur, L., and Alves, A. (2020). AIA-BDE: A corpus of FAQs in Portuguese and their variations. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5442–5449, Marseille, France. European Language Resources Association.

[Guo and Diab, 2012] Guo, W. and Diab, M. (2012). Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 864–872, Jeju Island, Korea. Association for Computational Linguistics.

[Hartmann, 2016] Hartmann, N. (2016). Solo queue at assin: Combinando abordagens tradicionais e emergentes. *Linguamática*, 8(2):59–64.

[He et al., 2015] He, H., Gimpel, K., and Lin, J. (2015). Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1586, Lisbon, Portugal. Association for Computational Linguistics.

[He and Lin, 2016] He, H. and Lin, J. (2016). Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, San Diego, California. Association for Computational Linguistics.

[Herlihy and Rudinger, 2021] Herlihy, C. and Rudinger, R. (2021). MedNLI is not immune: Natural language inference artifacts in the clinical domain. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1020–1027, Online. Association for Computational Linguistics.

[Issa et al., 2018] Issa, F., Damonte, M., Cohen, S. B., Yan, X., and Chang, Y. (2018). Abstract meaning representation for paraphrase detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 442–452. Association for Computational Linguistics.

[Jaccard, 1912] Jaccard, P. (1912). The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50.

[Jawahar et al., 2019] Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

[Ji and Eisenstein, 2013] Ji, Y. and Eisenstein, J. (2013). Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 891–896, Seattle, Washington, USA. Association for Computational Linguistics.

[Jiang and de Marneffe, 2019] Jiang, N. and de Marneffe, M.-C. (2019). Evaluating BERT for natural language inference: A case study on the CommitmentBank. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6086–6091, Hong Kong, China. Association for Computational Linguistics.

[Jiang et al., 2021] Jiang, Z., Zhang, Y., Yang, Z., Zhao, J., and Liu, K. (2021). Alignment rationale for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5372–5387, Online. Association for Computational Linguistics.

[Jimenez et al., 2014] Jimenez, S., Dueñas, G., Baquero, J., and Gelbukh, A. (2014). UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 732–742, Dublin, Ireland. Association for Computational Linguistics.

[Jurafsky and Martin, 2009] Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

[Kalouli et al., 2019] Kalouli, A.-L., Buis, A., Real, L., Palmer, M., and de Paiva, V. (2019). Explaining simple natural language inference. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 132–143, Florence, Italy. Association for Computational Linguistics.

[Kamp and Reyle, 1993] Kamp, H. and Reyle, U. (1993). *From Discourse to Logic - Introduction to Model theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42 of *Studies in linguistics and philosophy*. Springer.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[Kovatchev et al., 2018] Kovatchev, V., Martí, M. A., and Salamó, M. (2018). ETPC - a paraphrase identification corpus annotated with extended paraphrase typology and negation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

[Kozareva and Montoyo, 2006] Kozareva, Z. and Montoyo, A. (2006). Paraphrase identification on the basis of supervised machine learning techniques. In *Proceedings of the International Conference on Advances in Natural Language Processing*.

[Lai and Hockenmaier, 2014] Lai, A. and Hockenmaier, J. (2014). Illinois-LH: A denotational and distributional approach to semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 329–334, Dublin, Ireland. Association for Computational Linguistics.

[Lan et al., 2017] Lan, W., Qiu, S., He, H., and Xu, W. (2017). A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234. Association for Computational Linguistics.

[Lan and Xu, 2018] Lan, W. and Xu, W. (2018). Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[Landauer and Dumais, 1997] Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

[Levy and Goldberg, 2014] Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

[Li et al., 2004] Li, M., Chen, X., Li, X., Ma, B., and Vitányi, P. (2004). The similarity metric. *Information Theory, IEEE Transactions on*, 50(12).

[Li and Srikumar, 2016] Li, T. and Srikumar, V. (2016). Exploiting sentence similarities for better alignments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2193–2203, Austin, Texas. Association for Computational Linguistics.

[Liang et al., 2016] Liang, C., Paritosh, P., Rajendran, V., and Forbus, K. D. (2016). Learning paraphrase identification with structural alignment. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2859–2865. AAAI Press.

[Lin and Hovy, 2003] Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

[Lin and Och, 2004] Lin, C.-Y. and Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*.

[Liu et al., 2018] Liu, J., Cohen, S. B., and Lapata, M. (2018). Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 429–439, Melbourne, Australia. Association for Computational Linguistics.

[Liu et al., 2021] Liu, J., Cohen, S. B., Lapata, M., and Bos, J. (2021). Universal Discourse Representation Structure Parsing. *Computational Linguistics*, 47(2):445–476.

[Liu et al., 2019a] Liu, L., Yang, W., Rao, J., Tang, R., and Lin, J. (2019a). Incorporating contextual and syntactic structures improves semantic similarity modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1204–1209, Hong Kong, China. Association for Computational Linguistics.

[Liu et al., 2019b] Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019b). Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

[Liu et al., 2020] Liu, T., Wang, X., Lv, C., Zhen, R., and Fu, G. (2020). Sentence matching with syntax- and semantics-aware BERT. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3302–3312, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[Liu et al., 2019c] Liu, X., He, P., Chen, W., and Gao, J. (2019c). Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

[Madnani et al., 2012] Madnani, N., Tetreault, J., and Chodorow, M. (2012). Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 182–190, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Mamede et al., 2012] Mamede, N. J., Baptista, J., Diniz, C., and Cabarrão, V. (2012). STRING: An hybrid statistical and rule-based natural language processing chain for portuguese. *International Conference on Computational Processing of Portuguese (Propor 2012)*, Demo Session.

[Manhaeve et al., 2018] Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and De Raedt, L. (2018). Deepproblog: Neural probabilistic logic programming. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Marelli et al., 2014a] Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., and Zamparelli, R. (2014a). SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.

[Marelli et al., 2014b] Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014b). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 216–223, Reykjavik, Iceland. European Languages Resources Association (ELRA).

[Marques, 2015] Marques, R. (2015). Detecting contradictions in news quotations. Master's thesis, IST, University of Lisbon.

[Martínez-Gómez et al., 2017] Martínez-Gómez, P., Mineshima, K., Miyao, Y., and Bekki, D. (2017). On-demand injection of lexical knowledge for recognising textual entailment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 710–720, Valencia, Spain. Association for Computational Linguistics.

[Martins, 2011] Martins, B. (2011). A supervised machine learning approach for duplicate detection over gazetteer records. In *Proceedings of the International Conference on GeoSpatial Semantics*.

[May, 2016] May, J. (2016). SemEval-2016 task 8: Meaning representation parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1063–1073, San Diego, California. Association for Computational Linguistics.

[May and Priyadarshi, 2017] May, J. and Priyadarshi, J. (2017). SemEval-2017 task 9: Abstract Meaning Representation parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 536–545, Vancouver, Canada. Association for Computational Linguistics.

[McClendon et al., 2014] McClendon, J. L., Mack, N. A., and Hodges, L. F. (2014). The use of paraphrase identification in the retrieval of appropriate responses for script based conversational agents. In Eberle, W. and Boonthum-Denecke, C., editors, *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014, Pensacola Beach, Florida, May 21-23, 2014.* AAAI Press.

[McCoy et al., 2019] McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

[Mihalcea et al., 2006] Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the National Conference on Artificial Intelligence*.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[Minervini and Riedel, 2018] Minervini, P. and Riedel, S. (2018). Adversarially regularising neural NLI models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74, Brussels, Belgium. Association for Computational Linguistics.

[Misra et al., 2016] Misra, A., Ecker, B., and Walker, M. (2016). Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287. Association for Computational Linguistics.

[Moschitti, 2006] Moschitti, A. (2006). Making tree kernels practical for natural language learning. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–120, Trento, Italy. Association for Computational Linguistics.

[Mrkšić et al., 2016] Mrkšić, N., Ó Séaghdha, D., Thomson, B., Gašić, M., Rojas-Barahona, L. M., Su, P.-H., Vandyke, D., Wen, T.-H., and Young, S. (2016). Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

[Naik et al., 2018] Naik, A., Ravichander, A., Sadeh, N., Rose, C., and Neubig, G. (2018). Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[Oliveira Alves et al., 2016] Oliveira Alves, A., Rodrigues, R., and Gonçalo Oliveira, H. (2016). Asapp: Alinhamento semântico automático de palavras aplicado ao português. *Linguamática*, 8(2):43–58.

[Pado et al., 2009] Pado, S., Galley, M., Jurafsky, D., and Manning, C. D. (2009). Robust machine translation evaluation with entailment features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 297–305. Association for Computational Linguistics.

[Pakray et al., 2011] Pakray, P., Bandyopadhyay, S., and Gelbukh, A. (2011). Textual entailment using lexical and syntactic similarity. *International Journal of Artificial Intelligence and Applications*, 2(1).

[Papineni et al., 2002a] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002a). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

[Papineni et al., 2002b] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002b). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[Philips, 1990] Philips, L. (1990). Hanging on the metaphone. *Computer Language Magazine*, 7(12).

[Pinheiro et al., 2017] Pinheiro, A., Ferreira, R., Ferreira, M. A. D., Rolim, V. B., and Tenório, J. V. S. (2017). Statistical and semantic features to measure sentence similarity in portuguese. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 342–347.

[Pires et al., 2019] Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

[Platt, 2000] Platt, J. (2000). Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. In Smola, A., Bartlett, P., Schoelkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*, pages 61–74.

[Poliak, 2020] Poliak, A. (2020). A survey on recognizing textual entailment as an NLP evaluation. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 92–109, Online. Association for Computational Linguistics.

[Poliak et al., 2018] Poliak, A., Haldar, A., Rudinger, R., Hu, J. E., Pavlick, E., White, A. S., and Van Durme, B. (2018). Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.

[Potthast et al., 2010] Potthast, M., Stein, B., Barrón-Cedeño, A., and Rosso, P. (2010). An evaluation framework for plagiarism detection. In *Coling 2010: Posters*, pages 997–1005, Beijing, China. Coling 2010 Organizing Committee.

[Qiu et al., 2006] Qiu, L., Kan, M.-Y., and Chua, T.-S. (2006). Paraphrase recognition via dissimilarity significance classification. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 18–26, Sydney, Australia. Association for Computational Linguistics.

[Real et al., 2020] Real, L., Fonseca, E., and Gonçalo Oliveira, H. (2020). The assin 2 shared task: A quick overview. In Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., and Gonçalves, T., editors, *Computational Processing of the Portuguese Language*, pages 406–412, Cham. Springer International Publishing.

[Real et al., 2018] Real, L., Rodrigues, A., Vieira e Silva, A., Albiero, B., Thalenberg, B., Guide, B., Silva, C., de Oliveira Lima, G., Câmara, I. C. S., Stanojević, M., Souza, R., and de Paiva, V. (2018). Sick-br: A portuguese corpus for inference. In Villavicencio, A., Moreira, V., Abad, A., Caseli, H., Gamallo, P., Ramisch, C., Gonçalo Oliveira, H., and Paetzold, G. H., editors, *Computational Processing of the Portuguese Language*, pages 303–312, Cham. Springer International Publishing.

[Rieck and Wressnegger, 2016] Rieck, K. and Wressnegger, C. (2016). Harry: A tool for measuring string similarity. *J. Mach. Learn. Res.*, 17(1):258–262.

[Rocha and Lopes Cardoso, 2018] Rocha, G. and Lopes Cardoso, H. (2018). Recognizing textual entailment: Challenges in the portuguese language. *Information*, 9(4):76.

[Rocktäschel and Riedel, 2017] Rocktäschel, T. and Riedel, S. (2017). End-to-end differentiable proving. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Rodrigues et al., 2022] Rodrigues, H., Nyberg, E., and Coheur, L. (2022). Towards the benchmarking of question generation: Introducing the monserrate corpus. *Lang. Resour. Eval.*, 56(2):573–591.

[Rodrigues et al., 2019a] Rodrigues, R., Couto, P., and Rodrigues, I. (2019a). IPR: the semantic textual similarity and recognizing textual entailment systems. In Oliveira, H. G., Real, L., and Fonseca, E., editors, *Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, BA, Brazil, October 15, 2019*, volume 2583 of *CEUR Workshop Proceedings*, pages 39–48. CEUR-WS.org.

[Rodrigues et al., 2019b] Rodrigues, R. C., da Silva, J. R., de Castro, P. V. Q., da Silva, N. F. F., and da Silva Soares, A. (2019b). Multilingual transformer ensembles for portuguese natural language tasks. In Oliveira, H. G., Real, L., and Fonseca, E., editors, *Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, BA, Brazil, October 15, 2019*, volume 2583 of *CEUR Workshop Proceedings*, pages 27–38. CEUR-WS.org.

[Romanov and Shivade, 2018] Romanov, A. and Shivade, C. (2018). Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium. Association for Computational Linguistics.

[Rus et al., 2013] Rus, V., Lintean, M., Banjade, R., Niraula, N., and Stefanescu, D. (2013). Semilar: The semantic similarity toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 163–168, Sofia, Bulgaria. Association for Computational Linguistics.

[Rus et al., 2008] Rus, V., McCarthy, P., Lintean, M., McNamara, D., and Graesser, A. (2008). Paraphrase identification with lexico-syntactic graph subsumption. In *Proceedings of the 21th International Florida Artificial Intelligence Research Society Conference, FLAIRS-21*, Proceedings of the 21th International Florida Artificial Intelligence Research Society Conference, FLAIRS-21, pages 201–206. 21th International Florida Artificial Intelligence Research Society Conference, FLAIRS-21 ; Conference date: 15-05-2008 Through 17-05-2008.

[Sadrzadeh and Kartsaklis, 2016] Sadrzadeh, M. and Kartsaklis, D. (2016). Compositional distributional models of meaning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 1–4, Osaka, Japan. The COLING 2016 Organizing Committee.

[Santos et al., 2019] Santos, J., Alves, A., and Oliveira, H. G. (2019). Asapppy: a python framework for portuguese STS. In Oliveira, H. G., Real, L., and Fonseca, E., editors, *Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, BA, Brazil, October 15, 2019*, volume 2583 of *CEUR Workshop Proceedings*, pages 14–26. CEUR-WS.org.

[Scherrer, 2020] Scherrer, Y. (2020). TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6868–6873, Marseille, France. European Language Resources Association.

[Schneider et al., 2014] Schneider, N., Danchik, E., Dyer, C., and Smith, N. A. (2014). Discriminative lexical semantic segmentation with gaps: Running the MWE gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

[Sequeira et al., 2012] Sequeira, J., Gonçalves, T., and Quaresma, P. (2012). Semantic role labeling for portuguese – a preliminary approach –. In Caseli, H., Villavicencio, A., Teixeira, A., and Perdigão, F., editors, *Computational Processing of the Portuguese Language*, pages 193–203, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Shi and Huang, 2020] Shi, Z. and Huang, M. (2020). Robustness to modification with shared words in paraphrase identification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 164–171, Online. Association for Computational Linguistics.

[Shi et al., 2021] Shi, Z., Liu, H., and Zhu, X. (2021). Enhancing descriptive image captioning with natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 269–277, Online. Association for Computational Linguistics.

[Silva and Rigo, 2018] Silva, A. d. B. and Rigo, S. J. (2018). Enhancing brazilian portuguese textual entailment recognition with a hybrid approach. *J. Comput. Sci.*, 14(7):945–956.

[Silva et al., 2010] Silva, J., Branco, A., Castro, S., and Reis, R. (2010). Out-of-the-box robust parsing of portuguese. In Pardo, T. A. S., Branco, A., Klautau, A., Vieira, R., and de Lima, V. L. S., editors, *Computational Processing of the Portuguese Language*, pages 75–85, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Sinha et al., 2021] Sinha, K., Parthasarathi, P., Pineau, J., and Williams, A. (2021). UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online. Association for Computational Linguistics.

[Sjöblom et al., 2018] Sjöblom, E., Creutz, M., and Aulamo, M. (2018). Paraphrase detection on noisy subtitles in six languages. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 64–73, Brussels, Belgium. Association for Computational Linguistics.

[Snover et al., 2006] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*.

[Socher et al., 2011] Socher, R., Huang, E. H., Pennin, J., Manning, C. D., and Ng, A. Y. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 801–809. Curran Associates, Inc.

[Sokolova and Lapalme, 2009] Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

[Souza et al., 2020] Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: Pretrained bert models for brazilian portuguese. In Cerri, R. and Prati, R. C., editors, *Intelligent Systems*, pages 403–417, Cham. Springer International Publishing.

[Tai et al., 2015] Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

[Talhadas, 2013] Talhadas, R. (2013). Semantic role labelling in european portuguese. Master's thesis, Universidade do Algarve/FCHS, Faro, Portugal.

[Talman and Chatzikyriakidis, 2019] Talman, A. and Chatzikyriakidis, S. (2019). Testing the generalization power of neural network models across NLI benchmarks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 85–94, Florence, Italy. Association for Computational Linguistics.

[Tenney et al., 2019] Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

[Tsuchida and Ishikawa, 2011] Tsuchida, M. and Ishikawa, K. (2011). A method for recognizing textual entailment using lexical-level and sentence structure-level features. *In Proceedings of the Text Analysis Conference*.

[Ul-Qayyum and Wasif, 2012] Ul-Qayyum, Z. and Wasif, A. (2012). Paraphrase identification using semantic heuristic features. *Research Journal of Applied Sciences, Engineering and Tecnhology*, 4(22).

[van der Goot and van Noord, 2015a] van der Goot, R. and van Noord, G. (2015a). ROB: using semantic meaning to recognize paraphrases. In Cer, D. M., Jurgens, D., Nakov, P., and Zesch, T., editors, *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 40–44. The Association for Computer Linguistics.

[van der Goot and van Noord, 2015b] van der Goot, R. and van Noord, G. (2015b). ROB: Using semantic meaning to recognize paraphrases. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 40–44, Denver, Colorado. Association for Computational Linguistics.

[van Noord et al., 2018a] van Noord, R., Abzianidze, L., Haagsma, H., and Bos, J. (2018a). Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

[van Noord et al., 2018b] van Noord, R., Abzianidze, L., Toral, A., and Bos, J. (2018b). Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

[van Noord et al., 2019] van Noord, R., Toral, A., and Bos, J. (2019). Linguistic information in neural semantic parsing with multiple encoders. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, pages 24–31, Gothenburg, Sweden. Association for Computational Linguistics.

[van Noord et al., 2020] van Noord, R., Toral, A., and Bos, J. (2020). Character-level representations improve DRS-based semantic parsing even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4587–4603, Online. Association for Computational Linguistics.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

[Ventura et al., 2020] Ventura, M., Veiga, J., Coheur, L., and Gama, S. (2020). The b-subtle framework: Tailoring subtitles to your needs. *Lang. Resour. Eval.*, 54(4):1143–1159.

[Wan et al., 2006] Wan, S., Dras, M., Dale, R., and Paris, C. (2006). Using dependency-based features to take the 'para-farce' out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 131–138, Sydney, Australia.

[Wang et al., 2019a] Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2019a). Superglue: A stickier benchmark for general-purpose language understanding systems. In Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 3266–3280. Curran Associates, Inc.

[Wang et al., 2018] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

[Wang et al., 2021] Wang, H., Ma, F., Wang, Y., and Gao, J. (2021). Knowledge-guided paraphrase identification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 843–853, Punta Cana, Dominican Republic. Association for Computational Linguistics.

[Wang et al., 2019b] Wang, H., Sun, D., and Xing, E. P. (2019b). What if we simply swap the two text fragments? a straightforward yet effective way to test the robustness of methods to confounding signals in nature language inference tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7136–7143.

[Wang et al., 2020] Wang, W., Bi, B., Yan, M., Wu, C., Xia, J., Bao, Z., Peng, L., and Si, L. (2020). Structbert: Incorporating language structures into pre-training for deep language understanding. In *International Conference on Learning Representations*.

[Wang et al., 2016] Wang, Z., Mi, H., and Ittycheriah, A. (2016). Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349, Osaka, Japan. The COLING 2016 Organizing Committee.

[Welleck et al., 2019] Welleck, S., Weston, J., Szlam, A., and Cho, K. (2019). Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

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

[Winkler, 1990] Winkler, W. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *Proceedings of the Section on Survey Research Methods*.

[Winters et al., 2022] Winters, T., Marra, G., Manhaeve, R., and Raedt, L. D. (2022). Deepstochlog: Neural stochastic logic programming. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9):10090–10100.

[Wu, 2005] Wu, D. (2005). Recognizing paraphrases and textual entailment using inversion transduction grammars. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 25–30, Ann Arbor, Michigan. Association for Computational Linguistics.

[Xiong et al., 2021] Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., and Singh, V. (2021). Nyströmformer: A nyström-based algorithm for approximating self-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14138–14148.

[Xu et al., 2015] Xu, W., Callison-Burch, C., and Dolan, B. (2015). SemEval-2015 task 1: Paraphrase and semantic similarity in twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

[y M. Antònia Martí y Horacio Rodríguez, 2010] y M. Antònia Martí y Horacio Rodríguez, M. V. (2010). Paraphrase concept and typology. a linguistically based and computationally oriented approach. *Procesamiento del Lenguaje Natural*, 46(0):83–90.

[Yang et al., 2019a] Yang, Y., Zhang, Y., Tar, C., and Baldridge, J. (2019a). PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural*

Language Processing (EMNLP-IJCNLP), pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

[Yang et al., 2019b] Yang, Z., Zhu, C., and Chen, W. (2019b). Parameter-free sentence embedding via orthogonal basis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 638–648, Hong Kong, China. Association for Computational Linguistics.

[Yin et al., 2020] Yin, W., Rajani, N. F., Radev, D., Socher, R., and Xiong, C. (2020). Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8229–8239, Online. Association for Computational Linguistics.

[Yin and Schütze, 2015] Yin, W. and Schütze, H. (2015). Discriminative phrase embedding for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1368–1373, Denver, Colorado. Association for Computational Linguistics.

[Yin et al., 2016] Yin, W., Schütze, H., Xiang, B., and Zhou, B. (2016). ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272.

[Zadrozny and Elkan, 2002] Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 694–699, New York, NY, USA. Association for Computing Machinery.

[Zhang et al., 2019a] Zhang, S., Ma, X., Duh, K., and Van Durme, B. (2019a). AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.

[Zhang et al., 2019b] Zhang, Y., Baldridge, J., and He, L. (2019b). PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

[Zhang and Patrick, 2005] Zhang, Y. and Patrick, J. (2005). Paraphrase identification by text canonicalization. In *Proceedings of the Australasian Language Technology Workshop*.

[Zhang et al., 2019c] Zhang, Z., Wu, Y., Li, Z., and Zhao, H. (2019c). Explicit contextual semantics for text comprehension. In *Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33)*.

[Zhao et al., 2014] Zhao, J., Zhu, T., and Lan, M. (2014). ECNU: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 271–277, Dublin, Ireland. Association for Computational Linguistics.

[Zhiguo Wang, 2017] Zhiguo Wang, Wael Hamza, R. F. (2017). Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4144–4150.

UNIVERSIDADE DE ÉVORA
INSTITUTO DE INVESTIGAÇÃO
E FORMAÇÃO AVANÇADA

Contactos:
Universidade de Évora
Instituto de Investigação e Formação Avançada — IIFA
Palácio do Vimioso | Largo Marquês de Marialva, Apart. 94
7002 - 554 Évora | Portugal
Tel: (+351) 266 706 581
Fax: (+351) 266 744 677
email: iifa@uevora.pt