GABRIELA SOUZA DE MELO

WINOGRAD SCHEMAS IN PORTUGUESE

São Paulo
2019

GABRIELA SOUZA DE MELO

WINOGRAD SCHEMAS IN PORTUGUESE

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

São Paulo
2019

GABRIELA SOUZA DE MELO

WINOGRAD SCHEMAS IN PORTUGUESE

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de Concentração:
Engenharia de Computação

Orientador:
Fábio Gagliardi Cozman

São Paulo
2019

RESUMO

O Desafio de Winograd se tornou uma referência em tarefas de resposta textual automatizada e processamento de linguagem natural. O conjunto original de Esquemas de Winograd foi criado na língua inglesa; de forma a estimular o desenvolvimento do campo de Processamento de Linguagem Natural em português, desenvolvemos um conjunto de Esquemas de Winograd em português. Também adaptamos soluções propostas para a versão do desafio baseada em inglês, de forma a ter um patamar inicial para a versão do desafio baseada em português; para fazê-lo, criamos um modelo de linguagem baseado em um conjunto de documentos da Wikipedia. De forma a avaliar o impacto do aumento da capacidade do modelo de linguagem nos resultados para o desafio em português, nós testaremos o treinamento do modelo com métodos estado-da-arte, e compararemos esses novos resultados com aqueles que foram obtidos por soluções estado-da-arte para o desafio em inglês.

Palavras-Chave – Desafio de Winograd, Aprendizagem de Máquina, Inteligência Artificial, Processamento de Linguagem Natural, Aprendizado Profundo.

ABSTRACT

The Winograd Schema Challenge has become a common benchmark for question answering and natural language processing. The original set of Winograd Schemas was created in English; in order to stimulate the development of Natural Language Processing in Portuguese, we have developed a set of Winograd Schemas in Portuguese. We have also adapted solutions proposed for the English-based version of the challenge so as to have an initial baseline for its Portuguese-based version; to do so, we created a language model for Portuguese based on a set of Wikipedia documents. In order to evaluate the impact of the increase in the language model capacity in the results for the Portuguese challenge, we will test state-of-the-art approaches for training language models, and compare these new results with those that have been obtained by state-of-the-art solutions for the English challenge.

Keywords – Winograd Schema Challenge, Machine Learning, Artificial Intelligence, Natural Language Processing, Deep Learning.

LIST OF FIGURES

1 Our language model architecture 31

LIST OF TABLES

1 Main Results 37

2 English Results - Partial Scoring 38

3 Portuguese Names and Manual Fixes 39

CONTENTS

1 Introduction **13**

1.1 Justification 14

1.2 Objectives 14

2 Background **17**

2.1 The Winograd Schema Challenge 17

2.2 Machine Learning 19

2.2.1 Machine Learning and Natural Language Processing 19

2.2.2 Deep Learning 20

2.3 Language Models 21

2.3.1 N-Gram Models 22

2.3.2 Neural Networks-based Models 23

3 Related Work **25**

3.1 Solvers for the English-based Version of the Challenge 25

3.2 Robustness of Solvers 26

4 A Portuguese-based Winograd Schema Challenge **29**

4.1 Collection of Portuguese-based Schemas 29

4.2 A Baseline Solver for the Portuguese-based WSC 30

5 Experiments **33**

5.1 Performance Analysis 33

5.1.1 Scoring of Sentences 33

5.1.2 Metrics 34

5.1.3 Subsets of Schemas 35
5.1.4 Manual Corrections 35
5.2 Preliminary Results 37
5.2.1 Comparison with English Models 38
5.2.2 Impact of Translation of Names and Manual Fixes 38

6 Conclusion and Future Work **41**

References **43**

Appendix A – Example of Translated Winograd Schemas - With Portuguese Names **47**

1 INTRODUCTION

The Winograd Schema Challenge is a reading comprehension challenge proposed in 2011 [1]. To this day, no solution has surpassed a 72.2% rate of correct answers[1] [2].

The challenge consists of sentences containing ambiguity related to a coreference problem with the pronouns. Each sentence contains a pronoun that could refer to either of the two noun phrases previously mentioned in the sentence and is followed by a question about which of the noun phrases the pronoun refers to.

An example of a Winograd Schema Challenge question could be:

> ***The trophy*** *doesn't fit in* ***the brown suitcase*** *because* ***it*** *it's too big. What is too big?* [3]

The WSC has been advocated as an alternative to the Turing Test. This is due to the fact that it contains particularly difficult coreference resolution problems that can only be solved using commonsense knowledge. These questions are simple for humans: in a test run in 2015, humans displayed 92% accuracy [4]. But the questions are challenging for computers; standard coreference resolution solvers do not work well on the challenge [5]. Another difficulty is the fact that there are few examples of sentences that actually qualify as Winograd Schemas (based on the rules proposed by Levesque, Davis and Morgenstern (2012) [3]); solutions that depend on training with very large datasets of Schemas do not work.

Interest in the WSC has also emerged as an evaluation tool for Language Models (a type of model in the field of Natural Language Processing), such as in the work by Radford et al. (2019), where Winograd Schemas are used as benchmarks [6]. Pronoun resolution as needed in the WSC is very relevant to the development of Natural Language Processing (NLP). Other tasks, such as machine translation, also depend on resolving ambiguous sentences as displayed in the WSC [7].

[1] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html

In order to stimulate the development of NLP research in Portuguese, we have created a set of Winograd Schemas in Portuguese. This task is not as simple as translating each Schema in English word by word; there are several rules to consider when developing a Schema, and their effect changes from language to language. We have also developed a system for solving Winograd Schemas in Portuguese; this system should serve as an initial baseline for the task.

1.1 Justification

With the lack of benchmarks available for the Natural Language Processing field in Portuguese, the measurement and evaluation of work in the field for this language gets slowed down.

Therefore, our project developed a set of Winograd Schemas in Portuguese and replicated a solution for the English language for it. This is based on the hypothesis that the same sort of solution that has shown good results for the English version would also have good results when applied to a different language version of the challenge.

It is also worth noting that the Winograd Schema Challenge is a specific type of the Pronoun Disambiguation Problem, which is a problem in natural language processing [8], related to coreference resolution [9]. Hence, the ability to correctly solve Winograd Schema Challenge questions can be expanded further into other problems regarding language models and natural language processing, helping contribute to the hypothesis that having this collection available in Portuguese would be beneficial to the field of Natural Language Processing as a whole.

1.2 Objectives

Our main objective is to develop a Portuguese set of Winograd schemas and to investigate whether methods that have been successful for the English version of the challenge can also be successful in the Portuguese one.

The output of our work will be the Portuguese set of schemas and a trained model capable of answering these schemas. The outcome of this is that we will then establish a baseline for the Portuguese version of the challenge.

Therefore, we hope to impact not only with a Winograd Schema Challenge in the Portuguese language but also more broadly, in the development of the field of Natural

Language Processing in Portuguese. This work also provides an example for systematic translation of the Winograd Schema set to other languages and for comparing results of solvers for the challenge in different languages, making publicly available all the code that was developed for being able to do so. Consequently, we also hope to impact the replication of similar work for other languages.

2 BACKGROUND

In this chapter, the background topics related to our work are presented. These are mainly the Winograd Schema Challenge – what it is, why it was proposed, what makes a Winograd Schema; Machine Learning and how it relates to language problems; and lastly, Language Models.

2.1 The Winograd Schema Challenge

Coreference resolution problems are situations in Natural Language Processing where the machine needs to figure out which entity an expression (usually a pronoun) refers to [10]. In many cases, this can be done simply by looking at gender or number agreement. However, this is not enough for solving Winograd Schemas. Rahman and Ng (2012) call this type of coreference problem present in Winograd Schemas "complex cases of definite pronouns", in regards to this additional difficulty [11].

This challenge was proposed as a new Turing Test, in that it would be a way of assessing if a machine is thinking. The argument for this is that the only way of answering Winograd Schema Challenge questions is through the use of commonsense knowledge - there are no linguistic elements that would make it possible to solve the questions without the usage of previous knowledge. These questions are simple for humans: in a test run in 2015, humans displayed 92% accuracy [4]. But the questions are challenging for computers; standard coreference resolution solvers do not work well on the challenge [5]. Levesque, Davis and Morgenstern (2012) argue that it is exactly the ability to bring together background knowledge that we informally refer to as thinking [3]. Therefore, if a machine was able to correctly answer these schemas, it would be said to be thinking. Rahman and Ng (2012) agree with this proposal and states that "this is an easy task for a subject who can 'understand' natural language but a challenging task for one who can only make intelligent guesses" [11].

The main arguments for using this new test instead of the Turing Test are that the

latter involves the machine needing to pretend to be somebody and that a conversation might not be the best manner of evaluating a machine's intelligence. The Winograd Schema Challenge also has the advantage of not depending on a judge to decide whether the machine is intelligent or not [3].

A set of rules has been established for a sentence to be considered a Winograd Schema [3]. In short, the rules determine that:

1. The possible antecedents are noun phrases of the same gender;
2. A pronoun or possessive adjective refers to one of these antecedents, but is also of the correct type for the other possible antecedent;
3. Answer 0 is always the first party mentioned in the sentence, answer 1 is the second;
4. There is a *special word*, that, when changed to the *alternate word*, the sentence is still perfectly valid, but the answer switches;
5. A Schema cannot be too obvious, in the sense that a simple statistical check of whether the special word happens more frequently with one of the possible answers than the other must not be able to solve the Schema;
6. The sentences must not be too ambiguous, that is, fluent speakers of the language must be able to correctly answer the sentence without doubts.

If we take the following Winograd Schema as an example:

> ***Joan*** *made sure to thank* ***Susan*** *for all the help* ***she*** *had given. Who had given help?* [1]

The correct answer would be Susan. The special word in this case is *given*, which can be substituted by *received*, as in the example below:

> ***Joan*** *made sure to thank* ***Susan*** *for all the help* ***she*** *had received. Who had received help?* [2]

After this change, the correct answer now becomes Joan.

[1] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.html
[2] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.html

2.2 Machine Learning

Machine learning is a subset of the artificial intelligence field in which a computer can learn by itself — based on data provided to it — instead of having to be explicitly programmed to perform such task [12]. It was defined by Mitchell et al. (1997) as follows: "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." [13]

Developing a machine learning solution usually includes selecting a model (e.g. linear regression, decision tree, neural network, amongst many others) and then fitting the model to the dataset that is available. By doing this, the goal is that the model, with the adjusted parameters, learns to represent the problem as accurately as possible.

Machine Learning models are valuable in situations where a vast amount of data is available. In fact, a common trick for achieving better results from a machine learning model is to use more data to train the algorithm with, as this may improve the model's generalization capabilities.

In the remaining of this session, we will discuss how Machine Learning relates to Natural Language Processing and introduce Deep Learning — a type of model that has been very frequently used for Natural Language Processing and Language Models.

2.2.1 Machine Learning and Natural Language Processing

The Machine Learning field has an intersection with the Natural Language Processing field. Machine Learning, although not restricted to the use in NLP tasks, is one of the possible tools for performing these tasks. Natural Language Processing (NLP) is a field concerned with automatic methods for handling language-related tasks - these tasks can be simple, such as spell-checking or finding synonyms of words, but can also be much more intricate, such as machine translation, question answering, or coreference resolution; the latter being a superset of the task that is asked of the machine on the Winograd Schema Challenge.

We notice in the available literature that machine learning solutions for text problems are mostly applied to the English language. The Winograd Schema Challenge, for instance, has only been translated to Japanese, Chinese [3] and French [14]. The English

[3] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html

version of the challenge, on the other hand, has seen numerous recent publications just in the past two years [2, 6, 15–19]. This not only makes the field too focused on English, but also creates a vicious cycle, in that, the very fact that there is plenty of work in the literature regarding the English language makes it so that the entire field is able to build on top of each other's work and progress faster. Such collaboration is not very much present in the Portuguese language, and the lack of benchmarks for Natural Language Processing in Portuguese might be another contributing factor for that.

2.2.2 Deep Learning

A subset of Machine Learning models — and one that has been particularly useful for NLP tasks in recent years — are those that are based on Deep Learning. This is an area of machine learning comprised of Neural Networks, which are models that work with representation learning. These models' architecture consists of multiple layers, each of which corresponds to a different level of representation of the input data [20]. Each layer is responsible for multiplying the input data by a set of weights. Equation (2.1) demonstrates the operation happening at each of the units, where $\mathbf{x}$ is the input vector, $\mathbf{w}$ the weight vector and b a bias value that is also learned at each of the units.

$$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{x}^T \mathbf{w} + b. \tag{2.1}$$

The layers have multiple units, and each unit has a vector of weights. Therefore, the function applied at each layer can be seen as a matrix multiplication over the input matrix.

A non-linear transformation is applied to the output of each layer, before passing those values on to the next layer. Some of the common non-linear transformations are the ReLU, the logistic sigmoid and the hyperbolic tangent functions. The final layer is referred to as *output layer*, whereas the intermediary layers are called *hidden layers*. [21]

The parameters that the model learns from training data are the weights for each of the units. Neural Networks work by updating the weights at each of the units, based on the output obtained from the usage of the current weights. This update is based on the backpropagation algorithm, which builds on top of gradient descent. The basic idea is to calculate the derivative of the loss with respect to each of the parameters and subtract that value, multiplied by a learning rate, from the previous value of the parameter. The initial way of doing this was to update the weights at the end of a training epoch —

that is, after the full set of training data was passed into the model. There are variations of this, however, mainly, Stochastic Gradient Descent — where the weights get updated after each training observation passes through the model — and the mini-batch manner of training, where the training data gets split into batches, and the weight updates happen at the end of each batch.

Ever since Neural Networks first started being used, many improvements to their operation have been proposed. There are a variety of different optimization algorithms that are used to improve the search for the best weights for the network, different regularization techniques, and multiple types of layers and mechanisms with which to connect the layers. The choice of the number of layers, units per layers and values for the hyperparameters are also of great impact in the network's performance; hence there is usually effort put into trying to find the best values for these.

Many of the usages of Neural Networks and Deep Learning in NLP tasks are in mechanisms that serve as base for developing systems for NLP applications, for instance, in the now very well established Word Embeddings [22], and in the more recent Transformer-based architectures [23], which are based on attention mechanisms and have brought new advances to results in the NLP field.

2.3 Language Models

Language models are are used to predict the probability of the next word in a sentence, or, in other words, finding which word is the most probable one following a sequence of words. That also means that it is a type of model capable of calculating the probability of a sequence of words happening together. Language models can be useful in applications such as dialogue systems, machine translation or text summarization, for instance.

Take the following sentence as an example:

The Winograd Schema Challenge is difficult

The language model then would be able to calculate the probability of this sentence, which can be expressed by the following equation:

$$P(w_1, ..., w_m) = \prod_{i=1}^{i=m} P(w_i | w_1, ..., w_{i-1}), \tag{2.2}$$

where w_1 corresponds in this sentence to the word *The*, and w_m to the word *difficult*.

We can also express the probability of the word *difficult*, in that sentence, as

$$P(w_{difficult}|w_{The}, w_{Winograd}, w_{Schema}, w_{Challenge}, w_{is}). \tag{2.3}$$

If the model has a good performance, we would expect that probability to be higher than that of a random word, for instance, $P(w_{ocean}|w_{The}, w_{Winograd}, w_{Schema}, w_{Challenge}, w_{is})$.

If we had the sentence *The Winograd Schema Challenge is* and wanted to predict which word would be the next most probable one, from a fixed-size vocabulary, we could then predict the probability of each of the words in that vocabulary, following the equation above, and then select the one with the highest probability as the next word to complete that sentence. After the next word has been selected, we can keep going with that word selection process, and have a full text derived from this.

There are a few different ways of developing models that work as Language Models, and these will be introduced in the sections below.

2.3.1 N-Gram Models

The first language models that were developed followed an approach that is referred to as *n-gram models*. This approach is based on the method of approximating probabilities as the count of occurrences of words in a text. Using this method, Expression (2.3) could then be estimated as:

$$P(w_{difficult}|w_{The}, \cdots, w_{is}) \approx \frac{count(w_{The}, w_{Winograd}, w_{Schema}, w_{Challenge}, w_{is}, w_{difficult})}{count(w_{The}, w_{Winograd}, w_{Schema}, w_{Challenge}, w_{is})}. \tag{2.4}$$

However, in situations where your object of interest involves long sentences, it would get very difficult to do so if we always kept calculating the count of occurrence of the entire text up to the word we are trying to predict. From this difficulty came the idea behind the naming of this type of model — instead of using the full text to find the next most probable word, we would only consider the previous n words. That is, we would look at the words as *n-grams*, and track the counts of those.

Following the previous example, if we were to look at bigrams, we would then reduce that equation to:

$$P(w_{difficult}|w_{is}) \approx \frac{count(w_{is}, w_{difficult})}{count(w_{is})}. \tag{2.5}$$

If we were to use trigrams instead, it would then become:

$$P(w_{difficult}|w_{challenge}, w_{is}) \approx \frac{count(w_{challenge}, w_{is}, w_{difficult})}{count(w_{challenge}, w_{is})}. \tag{2.6}$$

The challenge with this model is to find the best value for n. If it is too small, it will not capture the full meaning related to the upcoming word. Make it too large on the other hand and the calculation of the n-grams counts of occurrences will result in a large and sparse matrix and will take an increasingly long time to get calculated. Moreover, the probability of any *n-gram* that has not occurred yet would be set to zero, making it so that the model cannot generalize to unseen *n-grams*.

2.3.2 Neural Networks-based Models

Recently, most of the successful language models have been those based on artificial neural networks. Their usage makes it possible to avoid the difficulties related to N-Gram models, mentioned above: the sparsity of the matrix of counts (an issue referred to as *curse of dimensionality*) and the lack of generalization (that is, the lack of ability of the model to set the probability of an unseen word or *n-gram* to some value other than zero). The initial approaches to the usage of neural networks for language modeling would use multilayer perceptrons, where the first layer would consist of a word embedding layer which would have its weights shared between the sentence's words and a hidden layer which would receive as input the concatenation of the sentence's word embeddings. There would then be an output layer calculating a softmax distribution of probabilities for all the words in the vocabulary [24].

However, this method has a restriction regarding the sentence length: it can always only receive the same number of words as input. To tackle this restriction of having to pass as input a fixed-length sentence, Recurrent Neural Networks (RNN) started being used for language modeling [25].

Recurrent Neural Networks are a type of Neural Network layer where some information can be persisted between subsequent input data. This information is sometimes referred to as cell state or cell memory. In the context of text data, this means that we can make the network pass forward some memory information between words.

Furthermore, in this type of network, instead of updating the weights of the layers after each input is fed to the model, the weights can be shared by a sequence of inputs. This means that we can have the same weights being applied to each word in a sentence, and the weights get updated afterward. The weight updates for RNNs happen following the Backpropagation Through Time algorithm — it works just as regular backpropagation regarding the calculation of the gradients, but the gradients of all words in the sentence get summed before being used for updating the weights.

There are many benefits of using RNNs for language modeling. The first one, as already stated, is the possibility of feeding it different sized input. Additionally, the usage of longer inputs has no effects on model size. Another significant benefit is the fact that calculations use information from many steps back in time. Lastly, the fact that weights get shared between time steps mean that features learned across different positions of text can get shared.

However, it is necessary to mention a few disadvantages of RNNs as compared to other models: recurrent computation is slow, and it is difficult to use information from many steps back, due to vanishing and exploding gradients.

There are different variants of Recurrent Neural Networks (sometimes referred to as gated RNNs), namely the LSTM layer [26] and the GRU layer [27]. These layers are better than vanilla RNNs at modeling long-term dependencies between each input observation. In other words, they provide for a way of reducing the impact of vanishing and exploding gradients. This is achieved due to these layers containing mechanisms for allowing them to choose what to store and what to forget. GRUs are more recent than LSTMs and were presented as a simplification of the LSTM layer type.

RNNs are used for language modeling in a similar way as the non-RNN-based neural network models for language modeling. That is, there is usually an input and an output layers which act as a mapping between the vocabulary size and a reduced-dimensionality embedding size. In between these layers are the RNN layers. These layers receive a sequence of words and store a cell state which helps pass forward some memory information on the previously seen words. The final output of each RNN layer then gets passed on to the following layer.

3 RELATED WORK

Since the proposal of the English-based Winograd Schema Challenge, there have been many attempts at building machines that would pass it [2, 5, 6, 11, 15–19, 28–30]. For the proposal of a baseline solver for the Portuguese version of the challenge, these approaches were researched and studied. This chapter will present the main characteristics of these proposals, and how they relate to each other.

Recently, a discussion on the evaluation of the WSC results has been raised [31]; we will also present this discussion, at the end of this chapter.

3.1 Solvers for the English-based Version of the Challenge

Even though there are not enough examples of Winograd Schemas in order for a machine learning model to be trained just from examples of answered Schemas, some of these solvers utilize machine learning, by designing the systems in a way that does not rely on the usage of those schemas as training data for the learning algorithm. Among these systems that have been developed to solve the WSC, a number of them are based on understanding how the sentences are structured, and from this, to use examples or rules to resolve the ambiguity. Some of these systems derive sets of features from the sentences and use them for training the model [11]. Others are based on linguistic tools for parsing sentences [18, 30] or fitting them into predicate schemas [5] and using these strategies on the Winograd sentence and on results of queries on search engines. Yet others are based on relevance theory and knowledge graphs [28] and some on logic rules based on correlation formulas [29].

Another type of solver leverages the fact that language models trained on vast amounts of language corpora indirectly incorporate commonsense knowledge when learning word relations. These are relatively recent solutions, mostly based on Deep Learning, using neural networks based on embeddings [15], siamese networks [16], language model net-

works [17], and transformer architectures such as GPT-2 [6] and BERT [2,19]. Traditional linguistic tools for extracting dependency graphs in the sentences are also employed by Ruan et al. (2019) [19], who use this extracted information to complete the traditional transformer model [23].

Some of the existing solutions apply to a subset of Schemas, restricting its usability on a more general pronoun resolution scenario [5, 28–30].

To deal with the fact that we currently have a relatively small number of instances of Winograd schemas, some of the proposals in the literature have developed their custom datasets to help with training. Rahman and Ng (2012) have developed a set of relaxed Winograd schemas, containing 941 sentence pairs [11]. Following that proposal, this dataset has also been used in other works [2, 5, 16, 19]. Trinh and Le (2018) [17] and Kocijan et al. (2019) [2] have also developed custom datasets, based on text corpora.

While not being able to surpass 60% of accuracy of the results for the initial years after the challenge was proposed, the more recent works have been rapidly increasing these results. There is a clear relationship between the beginning of the utilization of language models based on deep neural networks and the improvement in these results, with the current state of the art being based on a Transformer structure [23], with the usage of the BERT model [32].

3.2 Robustness of Solvers

Recently, new evaluation criteria for the challenge have been proposed by Trichelair et al. (2018) [31]. These criteria are based on dividing the data into two new sets, based on associativity and switchability characteristics of the sentences.

The switchable set consists of sentences that can have the antecedents switched; the sentence is still valid and the answer to it switches accordingly. Therefore, for this switchable subset, we can have the unswitched (original) sentences, and the switched ones.

The associative set consists of sentences where one of the antecedents relates more strongly to the special word than the other (although this is one of the rules for the Winograd Schema Challenge collection, there has not been a strong check on whether all sentences follow this rule, and therefore some can be considered as being associative). Thus, all sentences in the collection are divided into the associative and the non-associative subsets.

The idea behind it is that this allows for further insights into model performance and facilitates understanding robustness to slight variations in sentences.

4 A PORTUGUESE-BASED WINOGRAD SCHEMA CHALLENGE

Here, the main contributions of our work are presented. These are, namely, the construction of the Portuguese set of Schemas and the development of a baseline solver for it, and will be presented in the aforementioned order.

4.1 Collection of Portuguese-based Schemas

Our Portuguese-based collection of Schemas was developed following the rules proposed by Levesque, Davis and Morgenstern (2012) [3], mentioned in Section 2.1. To develop our set, we used as a base the set of 285 original English-based Schemas that are available online [1] and manually translated each of them. Our translated set is also available both in a JSON format and in a more visually pleasing, HTML format.[2] Note that a few additional tags are present in the JSON format, and these are described in Section 5.1.3.

Three native Portuguese speakers worked on translating the sentences, all of whom were familiar with the Winograd Schemas Challenge and the rules regarding the consideration of a sentence as a Winograd Schema. Each sentence was translated by one of the speakers and validated by the other two. For eight sentences we could not find a suitable translation, and hence these were discarded in the Portuguese set.

At first, we keep names as they were in the original sentences. Nevertheless, the fact that many of these names are not commonly found in Portuguese speaking countries might interfere with the task. Hence, we have developed an additional collection where names have been replaced by popular names in Brazil. The only restriction for these substitutions was that gender would be kept unchanged. Also, names of famous personalities that appear in the Schemas, such as Madonna or Shakespeare, were kept as in the original

[1] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.xml

[2] https://github.com/gabimelo/portuguese_wsc/blob/master/data/processed/portuguese_wsc.json and https://github.com/gabimelo/portuguese_wsc/blob/master/data/raw/portugues_wsc.html

collection. This set is also available in HTML and JSON formats.[3]

Additionally, it is worth noting that there are some significant differences between Brazilian Portuguese and Portuguese in other Portuguese-speaking countries, such as Portugal. We have generated a collection of Schemas in Brazilian Portuguese, and we have not evaluated how natural they sound to Portuguese speakers from other countries.

Some of the Schemas had to be adapted in relation to the gender of the noun phrases, in comparison to the original Schemas. For instance, consider the sentence:

The trophy doesn't fit into **the brown suitcase** because it is too large.

Its literal translation would be:

O troféu não cabe **na maleta** porque **ele** é muito grande.

In Portuguese, however, objects are not gender-neutral, and, in this case, *troféu* is of masculine gender, while *maleta* is of feminine gender. This would make the pronoun *ele* to be very easily resolved, given that it refers to a masculine object, and the only masculine object in the sentence is *troféu*. We adapted such sentences so that they would follow all the rules for being a Winograd Schema, as long as there was a plausible adaptation that would not change the meaning of the sentence. For instance, we produced:

A medalha não cabe **na maleta** porque **ela** é muito grande.

Given that *medalha* is of feminine gender, it now becomes possible for the pronoun to refer to either *medalha* or *maleta*.

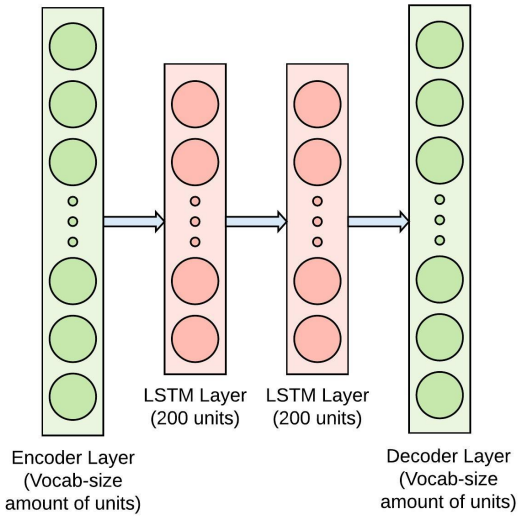
4.2 A Baseline Solver for the Portuguese-based WSC

To build a baseline solver for the Portuguese-based WSC we employed a Neural Network-based language model, as proposed by Trinh and Le (2018) [17]. Their solution is an ensemble of models that are so large that actually running a test is a nontrivial matter. We thus pursued a much simpler language model than they did, and also a single model instead of an ensemble of models, as we were mostly interested in establishing an initial baseline that other researchers can easily run.

[3]These datasets can be found at https://github.com/gabimelo/portuguese_wsc/blob/master/data/raw/portuguese_wsc_portuguese_names.html and https://github.com/gabimelo/portuguese_wsc/blob/master/data/processed/portuguese_wsc.json

For our language model, we used a neural network with input and output layers with hidden unit size equal to that of the vocabulary, acting as encoding and decoding layers, using an embedding size of 200, and two LSTM layers, with 200 hidden units each; as displayed in Figure 1. We used dropout between layers, with a probability of dropout equal to 0.2. The code for our solution has been made publicly available, in its entirety.[4]

Figure 1: Our language model architecture



The use of a language model for resolving Winograd Schemas goes as follows (the examples are in English but the process is the same for Portuguese schemas):

1. Each one of the candidate antecedents is substituted in place of the pronoun to be resolved. For instance, in this sentence:

   > The trophy doesn't fit into the brown suitcase because **it** is too large.

   We would then generate two sentences:

   > The trophy doesn't fit into the brown suitcase because **the trophy** is too large.
   >
   > The trophy doesn't fit into the brown suitcase because **the suitcase** is too large.

2. Each of these generated sentences is passed on to the model. In this step, all words in the sentence are sequentially sent to the language model; at each step, the generated probability for the next word in the sentence is stored.

[4] https://github.com/gabimelo/portuguese_wsc

3. The joint probability of each sentence is calculated. The sentence with the highest probability is assigned as the correct one. We employed two ways to calculate this probability, as did Trinh and Le (2018) [17]. We describe these two ways of scoring the sentences later in Section 5.1.1.

Our language model was trained using a corpus that we created for this task, as there seems to be no corpus currently established as a benchmark for language models in Portuguese. We derived our own from a Wikipedia dump which was the latest as of April 23rd, 2019 [5]. We used a subsection of the dump for training the model, equivalent to 15 MB (out of the original 2.3 GB), which contains 2,018,034 training tokens, 389,541 validation tokens, and 373,508 test tokens. The vocabulary consists of every word that appeared more than 5 times in the dataset, resulting in 32,032 unique tokens. Words outside of the vocabulary were replaced by an <unk> token and end of sentences were represented by <eos>. We have also made this corpus available.[6]

We also trained a similarly simple language model for English, and applied that to the original set of English Schemas, so as to compare how the approach works for these two different languages with similarly sized models. For this, the model was trained with the Wikitext-2 dataset [33]. This dataset contains 33,278 unique tokens, having 2,075,677 training tokens, 216,347 validation tokens, and 244,102 test tokens. This model follows the same architecture as the Portuguese model.

To train the models, we used learning rate annealing and started with an initial learning rate of 20. Input and output embeddings were tied, as proposed by both Press and Wolf (2017) and Inan, Khosravi and Socher (2016) [34, 35]. Gradients were clipped at 0.25 and training ran for 40 epochs. Sentences were organized into sequences of length 35.

[5] https://dumps.wikimedia.org/ptwiki/latest/ptwiki-latest-pages-articles.xml.bz2
[6] https://github.com/gabimelo/portuguese_wsc/tree/master/data/processed

5 EXPERIMENTS

We will now analyze the results obtained by our baseline solver. The first step for this is to examine the methods that were used for this analysis – the scoring of the sentences, the metrics that were applied, how the Schemas were subdivided into further subsets, and some manual corrections that were used on the translated sentences. Then we move on to the presentation of results, comparing both our solver applied to the Portuguese challenge and to the English challenge and the usage of our method applied to the English challenge as compared to the previous results presented on the literature for the English-based challenge [17]. We also provide results on the alternate versions of the dataset that we are providing: the one with Portuguese names and the one with manual corrections.

5.1 Performance Analysis

This section presents details on how the performance of our solver was evaluated. It starts by explaining the two different scoring methods we utilized. We then indicate the metrics used when presenting the results for our models. As we mentioned in Chapter 3, new subsets for the evaluation of WSC solvers have been proposed by Trichelair et al. (2018), and have, since then, been used for reporting the performance of such solvers [31]. Section 5.1.3 discusses the incorporation of this approach in our results. Lastly, we argue that grammatical mistakes introduced by the automatic substitution of candidate antecedents in place of the pronouns to be resolved might impact on the performance of solvers, and explain how we measured this impact.

5.1.1 Scoring of Sentences

There are two approaches proposed by Trinh and Le (2018) for the scoring of the sentences [17]. The first type is called full scoring. It is the ordinary joint probability of the sentence. That is:

$$Score_{full}(w_k \leftarrow c) = P_\theta(w_1, w_2, ..., w_{k-1}, c, w_{k+1}, ..., w_n).$$

where $w_k \leftarrow c$ indicates the word at position k being substituted by candidate c. The second way of scoring the model is with partial scoring, which is described as:

$$Score_{partial}(w_k \leftarrow c) = P_\theta(w_{k+1}, ..., w_n | w_1, ..., w_{k-1}, c).$$

We used the two approaches and present the results for each of them.

5.1.2 Metrics

Some of the works related to solving the Winograd Schema Challenge present their results in terms of the accuracy or precision metrics - accuracy being used when all of the WSC sentences are answered by the model, and precision when otherwise. Emami et al. (2018) argued that the F1 Score would be a more suitable metric when the solution being used presents answers for only some, but not all, of the sentences in the Winograd Schema set [18]. In these cases, they constructed the F1 Score by having the values for recall and precision being defined as:

$$recall = \frac{\#Correct}{Size\ of\ Winograd\ Set}\ ; \quad precision = \frac{\#Correct}{\#Answered}.$$

It can be noted that the definition of recall is the same as for the accuracy when the latter is calculated on the full set. Therefore, when the amount of answered Schemas is the same as the size of the full set, *recall* and *precision* both become the same, and in this case, based on the definition of the F1 Score, we can see that accuracy and F1 Score are equivalent.

$$F1 = 2 * \frac{precision * recall}{precision + recall}.$$

Given that this is the case for our system (all sentences can get answered by our model), for evaluating the results, we used as our metric the accuracy of the answers.

We also use the consistency metric, which calculates for how many of the switchable sentences the system switches the answer accordingly between the original sentence and the switched version of it.

5.1.3 Subsets of Schemas

Based on the criteria established by Trichelair et al. (2019) and mentioned in Chapter 3, we have used the subsets of the datasets made available by their work [31]. These subsets propose a way for better understanding the robustness of solvers and were called the switchable and the associative sets.

For the Portuguese set, we considered as associative and as switchable the same sentences that were marked as such on their work. This resulted in 35 associative sentences and 135 switchable sentences. We have two associative sentences less than the work in English (two of the sentences that were not able to be translated into Portuguese had been considered associative in the English collection).

Our collection of Schemas with names translated to Portuguese was also subdivided into these subsets.

5.1.4 Manual Corrections

When doing automatic substitutions of the pronouns, some of the cases become grammatically incorrect. For instance, in the sentence:

> **Jim** signaled **the barman** and gestured toward **his** empty glass.

Which has as pronoun to be resolved *his* and as possible answers *A. Jim* or *B. The barman*, the automatic substitutions becomes:

> Jim signaled the barman and gestured toward **Jim** empty glass.
> Jim signaled the barman and gestured toward **the barman** empty glass.

These sentences are missing the "'s" after the substituted candidates. We noticed that these sentences were left without any further corrections on the dataset of sentences after substitutions released by Trinh and Le (2018) and that there has not been any work analyzing if such errors might impact on the performance of WSC solvers [17].

In Portuguese, there are even more cases in which the automatic substitution does not work than in the English ones (where these cases are restricted to the usage of his/her pronouns). In Portuguese, the first type of sentence where this sort of error occurs are sentences with the usage of possessive pronouns is similar to the one presented in English, and can be exemplified in the following sentence:

> Há uma fenda na parede. É possível enxergar o jardim através **dela**.

Where the substitution for the first possible antecedent becomes:

> Há uma fenda na parede. É possível enxergar o jardim através **a fenda**.

When it should instead be:

> Há uma fenda na parede. É possível enxergar o jardim através **da fenda**.

Another case is that where the pronoun appears joined with the verb (a common Portuguese sentence construction):

> Eu estava tentando abrir o cadeado com a chave, mas alguém havia preenchido a fechadura com goma de mascar, e eu não conseguia **removê-la**.

Resulting in:

> Eu estava tentando abrir o cadeado com a chave, mas alguém havia preenchido a fechadura com goma de mascar, e eu não conseguia **removê-a goma de mascar**.

Instead of:

> Eu estava tentando abrir o cadeado com a chave, mas alguém havia preenchido a fechadura com goma de mascar, e eu não conseguia **remover a goma de mascar**.

The last case where there were issues with the automatic substitution were those where the pronoun appears before the verb:

> Eu usei um pano velho para limpar o alicate, e então **o coloquei** no lixo.

Resulting in:

> Eu usei um pano velho para limpar o alicate, e então **o pano coloquei** no lixo.

But when the pronoun gets substituted it should instead be:

> Eu usei um pano velho para limpar o alicate, e então **coloquei o pano** no lixo.

For the English-based collection of Schemas, we kept these substituted sentences unchanged — that is, in the same manner as they were used by previous work that utilized this set. This was done so in order to have accurate comparisons of results. For the Portuguese set, we developed, in addition to the collection of sentences after the automatic substitutions, a collection containing the manually fixed sentences, in order to assess whether this sort of treatment for the automatic substitution of the pronouns would make any difference. This helps understand how well the solution proposed here could be expanded to less structured pronoun resolution problems - where manual fixes might not be feasible.

5.2 Preliminary Results

Table 1 presents the main results from our experiments. Each of the rows, except for the last, represents the accuracy for a subset of the dataset, for each of the scoring techniques (full and partial). The last row refers to the consistency, which was defined in Section 5.1.2.

Table 1: Main Results

		English	Portuguese
Original (Full Dataset)	Full	50.55%	45.13%
	Partial	49.08%	44.77%
Associative	Full	45.95%	34.29%
	Partial	54.05%	51.43%
Non-Associative	Full	51.27%	46.69%
	Partial	48.3%	43.80%
Switched	Full	48.09%	40.74%
	Partial	51.14%	39.26%
Unswitched	Full	48.85%	42.22%
	Partial	46.56%	42.96%
Consistency	Full	4.58%	18.52%
	Partial	8.40%	28.15%

Using the same model with very similar vocabulary sizes for the English and Portuguese languages, the English language shows better results on the Winograd Schema Challenge. The fluctuation of results between each of the subsets was similar for the two

languages. Because the actual corpus used for training the model is of great influence on its performance, the fact that for the Portuguese model we used a corpus naively derived from the Wikipedia dump while for the English model we used a dataset that is a benchmark for language model work might be important, as the latter might have been more carefully crafted.

5.2.1 Comparison with English Models

It is important to compare the performance of our model for the English set of Winograd Schemas to that of the work from which we based our solution on [17]; this comparison is present in Table 2. Given that their publication was previous to that of the work introducing the associative and switchable subsets, we extracted the results from the latter [31]. We only compare the results from their single language model solution (their better solution involves an ensemble of multiple models). We also only compare to the partial scores, as the results from full scoring were not disclosed.

Table 2: English Results - Partial Scoring

	English - Single LM [31]	English - Ours
Original (Full Dataset)	54.58%	49.08%
Associative	73.0%	54.05%
Non-Associative	51.7%	48.30%
Switched	54.20%	51.14%
Unswitched	54.96%	46.56%
Consistency	56.49%	8.40%

In terms of model size, their model consists of almost 1.8 billion parameters, while ours has 7.3 million; the difference in the capacity of the models is very significant. These results show that an improved language model can perform substantially better in the English dataset [17], which leads us to believe that reaching better performance for the Portuguese collection might also be achieved through the improvement of the language model being used (our current Portuguese model is very similar in size to our English model, both consisting of 7 million parameters).

5.2.2 Impact of Translation of Names and Manual Fixes

As reported in Section 5.1.4, we made some manual fixes to the automatic substitution of candidate antecedents in place of pronouns, to analyze whether this would be a necessary measure when solving the WSC utilizing language models. Table 3 shows the

impact of manual fixes and also that of translating the names for some more commonly found in the Portuguese language.

Table 3: Portuguese Names and Manual Fixes

	Full	Partial
Portuguese	45.13%	44.77%
Portuguese - Manually Fixed	44.77%	45.13%
Portuguese - Portuguese Names	45.49%	44.04%
Portuguese - Portuguese Names - Manually Fixed	45.49%	44.77%

For our current approach, there was little difference in the result between the original set and the ones with these changes. The fact that manual corrections did not imply in a great difference in results suggests it might not be necessary to spend much effort into trying to improve the method for automatic substitution of candidates. Nevertheless, given that these aspects might have more of an influence if other approaches for solving the challenge were being used, we still find it relevant to release the collection of Portuguese Schemas with these translated names and manual fixes, in addition to the base collection.

6 CONCLUSION AND FUTURE WORK

We have developed a collection of Portuguese-based Winograd schemas; to the best of our knowledge, no similar collection exists today. The collection is of almost the same size as the original collection, as only a few sentences did not have a suitable translation found. This makes it so that we are now ready to have the Winograd Schema Challenge happening for the Portuguese language.

We have also created a baseline for solving the Portuguese-based WSC, based on an approach that has produced good results for the English-based version of the WSC [17]. The results obtained by this baseline show just how difficult it is to solve the Winograd Schema Challenge. It is worth noting that the solvers for the English challenge that had substantially larger performance than ours are all based on very large language models or linguistic models such as BERT [32]. This demonstrates that generic Natural Language Processing in Portuguese can benefit from such language models. Our baseline system provides for initial results that can be used as a means of comparison with future systems that attempt at tackling the challenge.

Our code is being made publicly available so that future researches working on expanding on our project will be able to reproduce our results and also use it as a starting point. The collection of schemas, as well as the indication of the subsets to which each schema belongs to, is also being made publicly available, as well as the text corpus that we have developed for training Portuguese based language models. An article has already been published at a national conference on the field.

Our work will focus on improving the current results that the model is able to obtain on the Portuguese version of the challenge. The main path of possible improvements in results that we intend to explore is changes and improvements to the language model in use. To do so we will work both on increasing model capacity, by increasing the size of the model being used, as well as improving the training process — by testing optimization algorithms, regularization, changes in amount of epochs, changes in the hyperparameters, and other tweaks available when working with neural networks, following state-of-the-art

methods. We will also run tests on changes in the training corpus that is being used to check whether that yields better results on the challenge.

1. Testing of changes in the training process.
2. Testing of different model sizes.
3. Testing of different corpus sizes.
4. Writing of Master's dissertation.
5. Presentation and deposit of Master's dissertation.

This is the expected schedule for the tasks listed above. The number for each task relates to the numbers on the list.

	Jan/20	Feb/20	Mar/20	Apr/20
1 - Training tests	X	X		
2 - Model tests	X	X		
3 - Corpus tests	X	X		
4 - Writing		X	X	
5 - Presentation				X

REFERENCES

[1] LEVESQUE, H. J. The Winograd schema challenge. In: *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*. [S.l.: s.n.], 2011. v. 46, p. 47.

[2] KOCIJAN, V. et al. A surprisingly robust trick for the Winograd schema challenge. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019. p. 4837–4842. Disponível em: <https://www.aclweb.org/anthology/P19-1478>.

[3] LEVESQUE, H. J.; DAVIS, E.; MORGENSTERN, L. The Winograd schema challenge. In: *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2012. (KR'12), p. 552–561. ISBN 978-1-57735-560-1. Disponível em: <http://dl.acm.org/citation.cfm?id=3031843.3031909>.

[4] BENDER, D. Establishing a human baseline for the Winograd schema challenge. In: *Proceedings of the 26th Modern AI and Cognitive Science Conference 2015, Greensboro, NC, USA, April 25-26, 2015*. [s.n.], 2015. p. 39–45. Disponível em: <http://ceur-ws.org/Vol-1353/paper_30.pdf>.

[5] PENG, H.; KHASHABI, D.; ROTH, D. Solving hard coreference problems. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, 2015. p. 809–819. Disponível em: <https://www.aclweb.org/anthology/N15-1082>.

[6] RADFORD, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog*, v. 1, n. 8, 2019.

[7] DAVIS, E. Winograd schemas and machine translation. *CoRR*, abs/1608.01884, 2016. Disponível em: <http://arxiv.org/abs/1608.01884>.

[8] MORGENSTERN, L.; DAVIS, E.; ORTIZ, C. L. Planning, executing, and evaluating the Winograd schema challenge. *AI Magazine*, v. 37, n. 1, p. 50–54, 2016.

[9] WEBSTER, K. et al. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, v. 6, p. 605–617, 2018. Disponível em: <https://www.aclweb.org/anthology/Q18-1042>.

[10] SOON, W. M.; NG, H. T.; LIM, D. C. Y. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, MIT Press, v. 27, n. 4, p. 521–544, 2001.

[11] RAHMAN, A.; NG, V. Resolving complex cases of definite pronouns: The Winograd schema challenge. In: *Proceedings of the 2012 Joint Conference on Empirical Methods*

in Natural Language Processing and Computational Natural Language Learning. Jeju Island, Korea: Association for Computational Linguistics, 2012. p. 777–789. Disponível em: <https://www.aclweb.org/anthology/D12-1071>.

[12] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, IBM, v. 3, n. 3, p. 210–229, 1959.

[13] MITCHELL, T. M. et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, v. 45, n. 37, p. 870–877, 1997.

[14] AMSILI, P.; SEMINCK, O. A Google-proof collection of French Winograd schemas. In: *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*. Valencia, Spain: Association for Computational Linguistics, 2017. p. 24–29. Disponível em: <https://www.aclweb.org/anthology/W17-1504>.

[15] LIU, Q. et al. Combing context and commonsense knowledge through neural networks for solving Winograd schema problems. *CoRR*, abs/1611.04146, 2016. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1611.htmlLiuJLZWH16>.

[16] OPITZ, J.; FRANK, A. Addressing the Winograd schema challenge as a sequence ranking task. In: *Proceedings of the First International Workshop on Language Cognition and Computational Models*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. p. 41–52. Disponível em: <https://www.aclweb.org/anthology/W18-4105>.

[17] TRINH, T. H.; LE, Q. V. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847, 2018. Disponível em: <http://arxiv.org/abs/1806.02847>.

[18] EMAMI, A. et al. A knowledge hunting framework for common sense reasoning. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 1949–1958. Disponível em: <https://www.aclweb.org/anthology/D18-1220>.

[19] RUAN, Y. et al. Exploring unsupervised pretraining and sentence structure modelling for Winograd schema challenge. *CoRR*, abs/1904.09705, 2019. Disponível em: <http://arxiv.org/abs/1904.09705>.

[20] LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.

[21] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. http://www.deeplearningbook.org.

[22] MIKOLOV, T. et al. Efficient estimation of word representations in vector space. In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. [s.n.], 2013. Disponível em: <http://arxiv.org/abs/1301.3781>.

[23] VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017. p. 5998–6008. Disponível em: <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.

[24] BENGIO, Y. et al. A neural probabilistic language model. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 1137–1155, mar. 2003. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=944919.944966>.

[25] MIKOLOV, T. et al. Recurrent neural network based language model. In: KOBAYASHI, T.; HIROSE, K.; NAKAMURA, S. (Ed.). *INTERSPEECH*. ISCA, 2010. p. 1045–1048. Disponível em: <http://dblp.uni-trier.de/db/conf/interspeech/interspeech2010.htmlMikolovKBCK10>.

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

[27] CHO, K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. Disponível em: <http://arxiv.org/abs/1406.1078>.

[28] SCHüLLER, P. Tackling Winograd schemas by formalizing relevance theory in knowledge graphs. In: *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2014. (KR'14), p. 358–367. ISBN 1-57735-657-8, 978-1-57735-657-8. Disponível em: <http://dl.acm.org/citation.cfm?id=3031929.3031973>.

[29] BAILEY, D. et al. The Winograd schema challenge and reasoning about correlation. In: *Working Notes of the Symposium on Logical Formalizations of Commonsense Reasoning*. AAAI Press, 2015. Disponível em: <http://www.cs.utexas.edu/users/ai-lab/?wsc15>.

[30] SHARMA, A. et al. Towards addressing the Winograd schema challenge: Building and using a semantic parser and a knowledge hunting module. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015. (IJCAI'15), p. 1319–1325. ISBN 978-1-57735-738-4. Disponível em: <http://dl.acm.org/citation.cfm?id=2832415.2832433>.

[31] TRICHELAIR, P. et al. On the evaluation of common-sense reasoning in natural language understanding. *CoRR*, abs/1811.01778, 2018. Disponível em: <http://arxiv.org/abs/1811.01778>.

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

[33] MERITY, S. et al. Pointer sentinel mixture models. *CoRR*, abs/1609.07843, 2016. Disponível em: <http://arxiv.org/abs/1609.07843>.

[34] PRESS, O.; WOLF, L. Using the output embedding to improve language models. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, 2017. p. 157–163. Disponível em: <https://www.aclweb.org/anthology/E17-2025>.

[35] INAN, H.; KHOSRAVI, K.; SOCHER, R. Tying word vectors and word classifiers: A loss framework for language modeling. *CoRR*, abs/1611.01462, 2016. Disponível em: <http://arxiv.org/abs/1611.01462>.

APPENDIX A – EXAMPLE OF TRANSLATED WINOGRAD SCHEMAS - WITH PORTUGUESE NAMES

1. Os vereadores recusaram a autorização aos manifestantes porque eles [temiam a/eram favoráveis à] violência.
   Answers: [Os vereadores/Os manifestantes]

2. A medalha não cabe na maleta porque ela é muito [grande/pequena].
   Answers: [A medalha/A maleta]

3. Jéssica certificou-se de agradecer Vanessa por toda ajuda que ela havia [recebido/oferecido].
   Answers: [Jéssica/Vanessa]

4. Paulo tentou ligar para o Gabriel, mas ele não [foi bem sucedido/estava disponível].
   Answers: [Paulo/Gabriel]

5. O advogado fez uma pergunta ao acusado, mas ele estava relutante em [repeti-la/respondê-la].
   Answers: [O advogado/O acusado]

6. O caminhão de entregas passou rapidamente pelo ônibus escolar porque ele estava indo muito [depressa/devagar].
   Answers: [O caminhão de entregas/O ônibus escolar]

7. Felipe sentiu-se [vingado/fracassado] quando seu rival de longa data André revelou que ele era o vencedor da competição.
   Answers: [Felipe/André]

8. O homem não conseguia erguer seu filho porque ele era muito [fraco/pesado].
   Answers: [O homem/O filho]

9. A grande bola atravessou pela mesa pois ela era feita de [aço/isopor].
   Answers: [A grande bola/A mesa]

10. João não conseguia enxergar o palco com Arthur na sua frente porque ele é muito [baixo/alto].
    Answers: [João/Arthur]

11. Vinícius jogou sua mochila para o Raimundo lá embaixo depois que ele atingiu [o topo/a parte inferior] da escada.
    Answers: [Vinícius/Raimundo]

12. Apesar de ambas correrem a aproximadamente a mesma velocidade, Sandra derrotou Marcia porque ela começou muito [bem/mal].
    Answers: [Sandra/Marcia]

13. A escultura caiu da prateleira porque ela não estava bem [acomodada/nivelada].
    Answers: [A escultura/A prateleira]

14. O desenho do Samuel estava pendurado imediatamente acima do da Tina e ele de fato ficava muito melhor com outro [abaixo/acima] dele.
    Answers: [O desenho do Samuel/O desenho da Tina]

15. Ana se saiu muito [melhor/pior] do que sua amiga Luciana na prova porque ela tinha estudado muito.
    Answers: [Ana/Luciana]

16. Os bombeiros chegaram [depois/antes] dos policiais porque eles estavam vindo de muito longe.
    Answers: [Os bombeiros/Os policiais]

17. Felipe estava chateado com o Vinícius porque a torradeira que ele havia [comprado dele/vendido para ele] não funcionava.
    Answers: [Felipe/Vinícius]

18. Guilherme [gritou com/reconfortou] Luiz porque ele estava muito transtornado.
    Answers: [Guilherme/Luiz]