LT2326 - Machine learning for statistical NLP: Advanced

Project – The Winograd Schema Challenge

The following report presents the work done for the project of the LT2326 course on Machine Learning for Statistical NLP: Advanced, by gusgraupa. The project is a very simple adaptation of the Winograd Schema Challenge, where we build a language model and get sentence probabilities.

First, there is a background of the project: a general idea of the Winograd Schema Challenge, as well as the type of sentence structure that it must follow. Second, the dataset that was used in this project is presented, both for the Winograd Schema, as well as for the data used to train the language model. Then, we can see the results of the model, as well as how to make it better and, finally, there is a conclusion explaining the biggest challenges of the project.

The implementation of the project can be found in this GitHub repository. The main file used is a Jupyter Notebook, which contains most of the code used, as well as a python file for some of the data pre-processing.

The Winograd Schema

The Winograd Schema Challenge or WSC started as a test for common-sense knowledge for artificial intelligence agents, introduced by Hector Levesque. It was designed as "another Turing test", and it contains sentences with a very specific structure. The following is a prototypical example of a Winograd Schema:

"The trophy $_x$ doesn't fit into the brown suitcase $_y$ because it $_z$ is too large."

"The trophy $_x$ doesn't fit into the brown suitcase $_y$ because it $_z$ is too small."

These two sentences are only different because of the last word: *large* or *small*. In the first sentence, when we say that "it is too large", the pronoun *it* is referring to *the trophy* (z=x). However, in the second one, what is "too small" is *the brown suitcase* (z=y). In the Winograd Schema Challenge, the objective is to identify the antecedent of this ambiguous pronoun.

There are different ways to do this, most of which are developed for English. For example, the work of Kocijan et al. (2020). This article presents two main ideas: first, the creation of a large dataset for the Winograd Schema Challenge that can be used to train a language model, and secondly, the training of BERT with that dataset. They use the masking of BERT to "determine which of the candidates is the correct replacement for the masked pronoun" (Kocijan et al., 2020, p. 2).

Nonetheless, there are approaches for other languages, such as the work of de Melo et al. (2020). In this article, the authors present the Winograd Schema Challenge for Portuguese. In this case, they manually translate the original dataset in English to Portuguese and create a Language Model, which contains two LSTM layers. They use the Language Model to encode two versions of each sentence of the original pair, where they replace the ambiguous pronoun for each of the possible referents. For this reason, the original two sentences that were previously presented, would become these four:

"The trophy doesn't fit into the brown suitcase because the trophy is too big."
"The trophy doesn't fit into the brown suitcase because the brown suitcase is too big."
"The trophy doesn't fit into the brown suitcase because the trophy is too small."
"The trophy doesn't fit into the brown suitcase because the brown suitcase is too small."

Then, they calculate the probability for each pair of sentences, and the sentence with the highest probability is assigned as the correct one. Both of these models, the one that is based on BERT and the one in Portuguese, are trained on Wikipedia articles, in the respective language.

The aim of the project was to create a dataset in Spanish for the Winograd Schema Challenge, and to follow one of these approaches. For the BERT approach, there are two resources available to use that have been pretrained in Spanish: there is Multilingual BERT, trained on multiple languages, or BETO, which has been pre-trained only in Spanish. However, mainly due to the small amount of data available for the task, I decided to follow the work of de Mela et al., and create a Language Model based on a LSTM.

Dataset

The dataset used in this project can be separated into two: first, the Winograd Schema sentences, from which we will calculate the sentence probability, and second, the data that was used to train the language model. The model was trained in two datasets, one based on Wikipedia articles in Spanish, and one based in sentences from different books, in the same language.

Winograd Schema Sentences

The Winograd Schema sentences were translated from the original 285 English-based Schemas that are available online (saved in *data/ English_data_full.txt*). The translation of the sentences had three steps. Firstly, they were passed in full to Google Translate. Secondly, they were manually corrected to follow correct and up-to-date usage of Spanish from Spain. Finally, they were modified to follow the structure of a Winograd Schema. Only 270 Winograd Schemas managed to get translated and be able to use for this project.

Since Spanish is a language with two genders, masculine and feminine, and the pronouns change depending on the gender of the antecedent, some of the referents of the sentences had to be changed, to be of the same gender. For instance, in the previous example, the Spanish word for *trophy*, 'trofeo', is masculine, whilst *suitcase*, 'maleta', is feminine. In this case, the first was changed to 'medalla', *medal*, because it is feminine. This way, both referents could be replaced by the same pronoun, in this case, 'esta'.

It must be briefly mentioned that most pronouns in subject position are dropped in Spanish, especially in the case where they have been explicitly mentioned previously. For the purpose of the project, however, they were kept explicit, creating some bizarre-sounding sentences, albeit correct.

Another issue that had to change most sentences was the verb *to be*. In English, there is only one verb to express this relation between the subject and the argument of the verb. However, there are two verbs in Spanish that can be used as the verb *to be*: 'ser' and 'estar'. Generally, the first represents a stable link between the subject and the argument of the verb (for example, in the sentence *he is tall*) and the latter, a weaker one (as in the example *he is tired*, where *tired* is only a state and not a trait). This is only a generalisation of its usage, as there are also many other structures which need to have one verb or the other. Some sentences in the original Winograd Schema dataset in English used the verb to be, like the examples provided. In the translation of the sentences, though, they had to change to accommodate this duality of the verb. Similarly, many sentences were changed because of constructions with prepositions in English, which do not work in Spanish for this kind of sentences.

The proper names that appeared in the sentences were changed to Spanish names, the appearance of which was guaranteed in the dataset for the language model. Finally, all disputed

pronouns were surrounded by underscores, since they were not the only time they appeared in the sentence, to avoid mistakes in the pre-processing of the sentences.

These sentences were processed in the function *sentence_creation*, from the notebook. From each sentence, this function creates two sentences, replacing the pronoun with either of the possible options. Then, each one is saved in a tuple with the Boolean True, if the pronoun has been correctly replaced, or False, if it was wrongly replaced. In the case of possessive forms (such as *his shoulders*), which follow the structure *noun of noun* when explicating the referent in Spanish, the pronoun was deleted, and the antecedent was placed after the noun (in the example, the output would be *shoulders of X*). The resulting sentences are missing the article before the noun (*the shoulders of X*), because there was no way of determining its gender. All sentences were lowercased.

Model dataset

There were two datasets that were used to train two different versions of the model. The first one comes from the Wikipedia, following the previously mentioned work. The second one is a dataset formed by 24 books picked at random from Project Gutenberg.

The Wikipedia dataset comes from the WikiCorpus in Spanish, which contains 120 million words. To process the data, we need to run the file *processing_wiki_data.py*. The function in this file separates the sentences, and eliminates some recurring patterns, as well as numbers and punctuation. The reason for doing this is trying to create a smaller vocabulary size, since the GPU kept getting out-of-memory errors when training the model. Again, all the sentences are lowercased, and only sentences with a size bigger than 8 are kept. The motivation behind this is the fact that the Wikipedia file contains many image descriptions and other short noun phrases, which I decided not to keep in the dataset. Since the shortest length of the Winograd Schema sentences is 9, only sentences with that size or bigger are kept. The function in the file *processing_wiki_data.py* creates the file *wiki_sent.csv*, which had to be truncated into the file *medium_wiki_sent.csv* due to its large size.

After a manual inspection of the resulting file, I saw that it contained many sentences in other languages (English, Latin, French, Catalan…), due to the nature of the articles of the Wikipedia. Because of that, I decided to create another dataset to train the language model, to get a smaller vocabulary size, and to check if the output of the model was better. This other dataset comes from 24 books of the Project Gutenberg, in Spanish. These books were picked at random, but those that contained theatre plays or poems were disregarded. The list of the books can be found in the file *books_names.txt*. It contains 59213 sentences, and it is considerably smaller than the previous dataset, which contains 100000 sentences. The data from the books was saved into the file *data/prova.py* and pre-processed in the notebook to the file *sentences.csv*.

The datasets were processed in the notebook with the function *get_data*, which creates and iterable and returns the vocabulary of each dataset. The 100000-line version of the dataset from Wikipedia contained 109829 words, whilst the one from the books contains 61781 unique words. The data was separated into batches of size 8, and a start and end token were added.

The Model

The language model was created with torch.nn and it contains an embedding layer, one unidirectional LSTM and two fully connected layers, as well as final softmax layer. The embedding layer comes from the size to the vocabulary, which is 109829 in the first model and 61781 in the second, to an embedding size of 256. Then, the LSTM layer brings this to a size of 128, and the linear layers

give an output of the size of the vocabulary. Finally, the output is passed through the softmax layer in the last dimension.

The reason behind this structure was to follow previous language models created for this course, but with the addition of some changes to try to make it better. For example, I decided to have two linear layers instead of one, hoping that the model get better results. The model was trained with the *train* function, using cross entropy loss as the loss function and Adam as the optimizer, passing through the data as described in the previous section, with batches of size 8.

The models were trained for 5 epochs on the GPU, and they were saved as *lstm_model_medium.pt* for the model created with the Wikipedia corpus, and *lstm_model_book.pt* for the model created with the corpus coming from the books. These models are not in the GitHub repository, but they can be found in the folder scratch/gusgraupa.

Results

The results were calculated with the function *getting_prob*, which passes the Winograd Schema sentences created in the function *sentence_creation* through the model and creates sentence probability. To get the sentence probabilities, the sentences were tokenized, encoded, and passed through both of the models. Then, they went through a softmax layer in the last dimension, where the element with the highest probability was taken. Afterwards, these word probabilities were multiplied to get the sentence probability.

In the case of the language model created with the Wikipedia dataset, we can see that only 40,74% of the sentences with the correct antecedent had a higher probability than the *incorrect* sentences, which contain the incorrect referent replacing the pronoun. The language model trained with the book corpus had a slightly better result, where 44,07% of the sentences had a higher probability. In both cases, these models perform worse or dangerously close to a random classifier.

One of the possible reasons for the poor results that we get from the language model is the dataset used. The first dataset, although it contains more data than the second one, it has worse quality. As mentioned, it contains words, and even full sentences, from other languages. This is due to the nature of the dataset, since Wikipedia articles are academic and contain not only quotes in other languages, but also words in Latin. In addition, the memory errors from the GPU made it impossible to use the whole available Wikipedia file for training, which contains 170107 sentences after being pre-processed. That is why I had to use a smaller version of the file, which contains 100000 sentences of the original file. Perhaps a different pre-processing of the data would also yield better results, such as the lemmatisation of the words.

The language model contains a unidirectional LSTM layer and two fully connected layers. A possible way to improve the model could be to make a bidirectional LSTM, since it has reached great results compared to other language models, such as BERT. Another idea could be to add another LSTM layer, following the work for the Portuguese Winograd Schema that de Melo et al. proposed in their paper (2020). Changing the sizes of the embedding size and hidden sizes in the model might also yield different results. For instance, the embedding layer of the model had a very big input size, of the size of the vocabulary, and a rather small output size, of 256. Another way of changing the model would be to change the hyperparameters of the training function, perhaps lowering the learning rate or training for more epochs.

Conclusion

This report has presented the work done for the project of the course LT2326, Machine Learning for NLP: Advanced, which is a very simple take on the Winograd Schema Challenge. The results of the model are quite bad, getting higher probabilities for the expected sentences only 44% of the cases, in the best performing model. Further work on how to better the has been presented in the previous section, which can be summarised into two approaches: using a different dataset, by means of changing the current ones or finding a new one that is better suited for the task, or changing the structure of the model.

Without a doubt, the most time-consuming part of this project has been getting the data and modifying it to suit the challenge. On one hand, creating the sentences for the Winograd Schema Challenge has been very laborious, not only because of the translating and changes that had to be made to accommodate the sentences to Spanish, but also because of the data structure. For example, when deciding how to keep the structure of the file. At first, I decided to keep the referents of the pronouns exactly how they appeared in the sentence. However, this meant that the sentences created when replacing the pronoun would be ungrammatical. For this reason, I had to manually check and rewrite all possible referents of the pronouns for each sentence.

Another example concerns how the pronouns appear in the sentence. As I commented in the section “Winograd Schema sentences”, the disputed pronouns were surrounded by underscores, because they occurred multiple times, in some sentences. This only became apparent after having processed the WS sentences, which meant having to change the data once again.

In fact, even the planification of the task in this project was difficult. Since Spanish is a language in which pronouns in subject position are not obligatory, at the beginning the task was thought as a question-answering problem. However, after seeing the proposal of Kocijan and de Melo, the task was adapted to sentence probability calculation.

The dataset from the Wikipedia was also very tedious to process, given all the image descriptions, mathematical functions, numbers and transcriptions, as well as many other characters that appeared in the original texts. In fact, after many changes, there are still some sentences that are not typical of natural language, such a sentence made of twenty letters separated by whitespaces. This is another of the reasons why I decided to create another dataset, as well as the sentences in other languages.

Acknowledgments

Thank you to guscasaju for helping me in the translation of some of the Winograd Schema Sentences to Spanish, whose help allowed me to get more sentences for the data for the model, which was the main objective in the case of working with BERT.

References

Kocijan, V., Cretu, A. M., Camburu, O. M., Yordanov, Y., & Lukasiewicz, T. (2020). A surprisingly robust trick for the winograd schema challenge. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics*, 4837–4842. https://doi.org/10.18653/v1/p19-1478

Melo, G., Imaizumi, V., & Cozman, F. (2020). *Winograd Schemas in Portuguese*. 787–798. https://doi.org/10.5753/eniac.2019.9334