BERT Masked Language Modeling for Co-reference Resolution

Felipe Alfaro Lois **José A. R. Fonollosa** **Marta R. Costa-jussà**
TALP Research Center
Universitat Politècnica de Catalunya, Barcelona
felipe.alfaro@est.fib.upc.edu
jose.fonollosa,marta.ruiz}@upc.edu

Abstract

This paper explains the TALP-UPC participation for the Gendered Pronoun Resolution shared-task of the 1st ACL Workshop on Gender Bias for Natural Language Processing. We have implemented two models for mask language modeling using pre-trained BERT adjusted to work for a classification problem. The proposed solutions are based on the word probabilities of the original BERT model, but using common English names to replace the original test names.

1 Introduction

The Gendered Pronoun Resolution task is a natural language processing task whose objective is to build pronoun resolution systems that identify the correct name a pronoun refers to. It's called a co-reference resolution task. Co-reference resolution tackles the problem of different elements of a text that refer to the same thing. Like for example a pronoun and a noun, or multiple nouns that describe the same entity. There are multiple deep learning approaches to this problem. NeuralCoref [1] presents one based on giving every pair of mentions (pronoun + noun) a score to represent whether or not they refer to the same entity. In our current task, this approach is not possible, because we don't have the true information of every pair of mentions, only the two names per entry.

The current task also has to deal with the problem of gender. As the GAP researchers point out (Webster et al., 2018), the biggest and most common datasets for co-reference resolution have a bias towards male entities. For example the OntoNotes dataset, which is used for some of the most popular models, only has a 25% female representation (Pradhan and Xue, 2009). This creates a problem, because any machine learning model is only as good as its training set. Biased training sets will create biased models, and this will have repercussions on any uses the model may have.

This task provides an interesting challenge specially by the fact that it is proposed over a gender neutral dataset. In this sense, the challenge is oriented towards proposing methods that are gender-neutral and to not provide bias given that the data set does not have it.

To face this task, we propose to make use of the recent popular BERT tool (Devlin et al., 2018). BERT is a model trained for masked language modeling (LM) word prediction and sentence prediction using the transformer network (Vaswani et al., 2017). BERT also provides a group of pre-trained models for different uses, of different languages and sizes. There are implementations for it in all sorts of tasks, including text classification, question answering, multiple choice question answering, sentence tagging, among others. BERT is gaining popularity quickly in language tasks, but before this shared-task appeared, we had no awareness of its implementation in co-reference resolution. For this task, we've used an implementation that takes advantage of the masked LM which BERT is trained for and uses it for a kind of task BERT is not specifically designed for.

In this paper, we are detailing our shared-task participation, which basically includes descriptions on the use we gave to the BERT model and on our technique of 'Name Replacement' that allowed to reduce the impact of name frequency.

2 Co-reference Resolution System Description

2.1 BERT for Masked LM

This model's main objective is to predict a word that has been masked in a sentence. For this exer-

[1] https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30

Proceedings of the 1st Workshop on Gender Bias in Natural Language Processing, pages 76–81
Florence, Italy, August 2, 2019. ©2019 Association for Computational Linguistics

cise that word is the pronoun whose referent we're trying to identify. This one pronoun gets replaced by the *[MASKED]* tag, the rest of the sentence is subjected to the different name change rules described in section 2.2.

The text is passed through the pre-trained BERT model. This model keeps all of its weights intact, the only changes made in training are to the network outside of the BERT model. The resulting sequence then passes through what is called the masked language modeling head. This consists of a small neural network that returns, for every word in the sequence, an array the size of the entire vocabulary with the probability for every word. The array for our masked pronoun is extracted and then from that array, we get the probabilities of three different words. These three words are : the first replaced name (name 1), the second replaced name (name 2) and the word *none* for the case of having none.

This third case is the strangest one, because the word *none* would logically not appear in the sentence. Tests were made with the original pronoun as the third option instead. But the results ended up being very similar albeit slightly worse, so the word none was kept instead. These cases where there is no true answer are the hardest ones for both of the models.

We experimented with two models.

Model 1 After the probabilities for each word are extracted, the rest is treated as a classification problem. An array is created with the probabilities of the 2 names and *none* (*[name 1, name 2, none]*), where each one represents the probability of a class in multi-class classification. This array is passed through a softmax function to adjust it to probabilities between 0 and 1 and then the log loss is calculated. A block diagram of this model can be seen in figure 1.

Model 2 This model repeats the steps of model 1 but for two different texts. These texts are mostly the same except the replacement names *name 1* and *name* 2 have been switched (as explained in the section 2.2). It calculates the probabilities for each word for each text and then takes an average of both. Then finally applies the softmax and calculates the loss with the average probability of each class across both texts. A block diagram of this model can be seen in figure 2.

2.2 Name Replacement

The task contains names of individuals who are featured in Wikipedia, and some of these names are uncommon in the English language. As part of the pre-processing for both models, these names are replaced. They are replaced with common English names in their respective genders[2]. If the pronoun is female, one of two common English female names are chosen, same thing for the male pronouns. In order to replace them in the text, the following set of rules are followed.

1. The names mentioned on the A and B columns are replaced.
2. Any other instances of the full name as it appears on the A/B columns are replaced.
3. If the name on the A/B column contains a first name and a last name. Instances of the first name are also replaced. Unless both entities share a first name, or the first name of one is contained within the other.
4. Both the name and the text are converted to lowercase

This name replacement has two major benefits. First, the more common male and female names work better with BERT because they appear more in the corpus in which it is trained on. Secondly, when the word piece encoding splits certain words the tokenizer can be configured so that our chosen names are never split. So they are single tokens (and not multiple word pieces), which helps the way the model is implemented.

Both models (1 and 2 presented in the above section) use BERT for Masked LM prediction where the mask always covers a pronoun, and because the pronoun is a single token (not split into word pieces), it's more useful to compare the masked pronoun to both names, which are also both single tokens (not multiple word pieces).

Because the chosen names are very common in the English language, BERT's previous training might contain biases towards one name or the other. This can be detrimental to this model where it has to compare between only 3 options. So the alternative is the approach in model number 2. In model 2 two texts are created. Both texts are basically the same except the names chosen as the

[2] https://www.ef.com/wwen/english-resources/english-names/

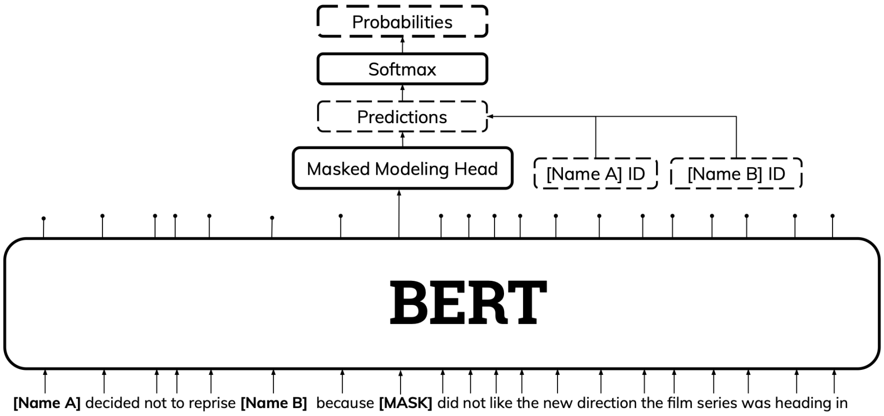

Probabilities
Softmax
Predictions
Masked Modeling Head
[Name A] ID
[Name B] ID
BERT
[Name A] decided not to reprise [Name B] because [MASK] did not like the new direction the film series was heading in

Figure 1: Model 1 representation.

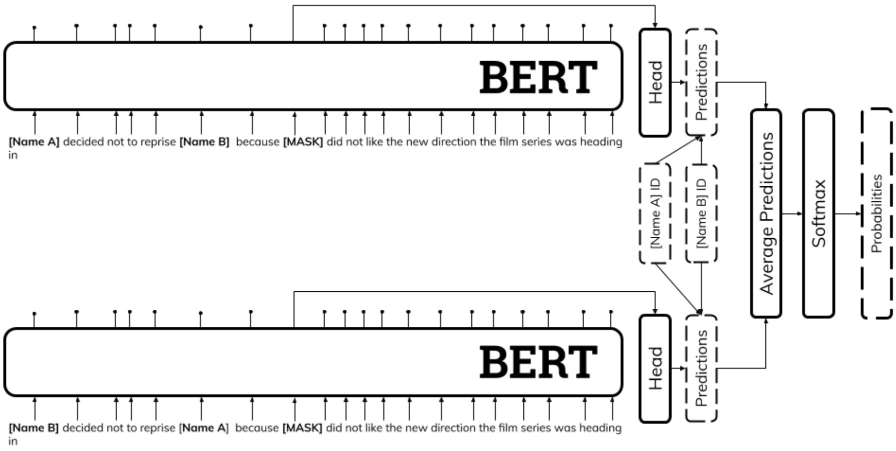
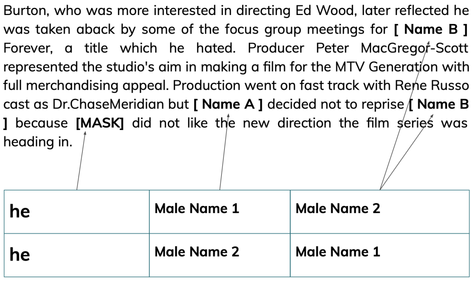

BERT
Head
Predictions
[Name A] decided not to reprise [Name B] because [MASK] did not like the new direction the film series was heading in
[Name A] ID
[Name B] ID
Average Predictions
Softmax
Probabilities
BERT
Head
Predictions
[Name B] decided not to reprise [Name A] because [MASK] did not like the new direction the film series was heading in

Figure 2: Model 2 representation.



Burton, who was more interested in directing Ed Wood, later reflected he was taken aback by some of the focus group meetings for **[Name B]** Forever, a title which he hated. Producer Peter MacGregor-Scott represented the studio's aim in making a film for the MTV Generation with full merchandising appeal. Production went on fast track with Rene Russo cast as Dr.ChaseMeridian but **[Name A]** decided not to reprise **[Name B]** because **[MASK]** did not like the new direction the film series was heading in.

he	Male Name 1	Male Name 2
he	Male Name 2	Male Name 1

Figure 3: Example of a text present in the dataset and how the word replacement was done for the model 2.

replacement names 1 and 2 are switched. So, as figure 3 shows, we get one text with each name in each position.

For example lets say we get the text:

”In the late 1980s Jones began working with Duran Duran on their live shows and then in the studio producing a B side single “This Is How A Road Gets Made”, before being hired to record the album Liberty with producer Chris Kimsey.”,

A is *Jones* and B is *Chris Kimsey*. For the name replacement lets say we choose two common English names like **John** and **Harry**. The new text produced for model 1 (figure 1) would be something like:

*”in the late 1980s **harry** began working with duran duran on their live shows and then in the studio producing a b side single “this is how a road gets made”, before being hired to record the album liberty with producer **john**.”*

And for model 2 (figure 2) the same text would be used for the top side and for the bottom side it would have the harry and john in the opposite positions.

3 Experimental Framework

3.1 Task details

The objective of the task is that of a classification problem. Where the output for every entry is the probability of the pronoun referencing name A, name B or Neither.

3.2 Data

The GAP dataset (Webster et al., 2018) created by Google AI Language was the dataset used for this task. This dataset consists of 8908 co-reference labeled pairs sampled from Wikipedia, also it's split perfectly between male and female representation. Each entry of the dataset consists of a short text, a pronoun that is present in the text and its offset and two different names (name A and name B) also present in the text. The pronoun refers to one of these two names and in some cases, none of them. The GAP dataset doesn't contain any neutral pronouns such as *it* or *they*.
For the two different stages of the competition different datasets were used.

- For **Stage 1** the data used for the submission is the same as the development set available in the GAP repository. The dataset used for training is the combination of the GAP validation and GAP testing sets from the repository.

- For **Stage 2** the data used for submission was only available through Kaggle[3] and the correct labels have yet to be released, so we can only analyze the final log loss of each of the models. This testing set has a total of 12359 rows, with 6499 male pronouns and 5860 female ones. For training, a combination of the GAP development, testing and validation sets was used. And, as all the GAP data, it is evenly distributed between genders.

The distributions of all the datasets are shown in table 1. It can be seen that in all cases, the *None* option has the least support by a large margin. This, added to the fact that the model naturally is better suited to identifying names rather than the absence of them, had a negative effect on the results.

	Stage 1		**Stage 2**
	Train	**Test**	**Train**
Name A	1105	874	1979
Name B	1060	925	1985
None	289	201	490

Table 1: Dataset distribution for the datasets of stages 1 and 2.

3.3 Training details

For the BERT pre-trained weights, several models were tested. BERT base is the one that produced the best results. BERT large had great results in a lot of other implementations, but in this model it produced worse results while consuming much more resources and having a longer training time. During the experiments the model had an overfitting problem, so the learning rate was tuned as well as a warm up percentage was introduced. As table 2 shows, the optimal learning rate was $3e-5$ while the optimal with a 20% warm up. The length of the sequences is set at 256, where it fits almost every text without issues. For texts too big, the text is truncated depending on the offsets of each of the elements in order to not eliminate any of the names or the pronoun.

		Accuracy		**Loss**	
Learning Rate	**Warmup**	**mean**	**min**	**mean**	**min**
0.00003	**0.0**	0.840167	0.8315	0.519565	0.454253
	0.2	0.844444	0.8340	0.502667	0.442313
0.00004	**0.0**	0.822389	0.7970	0.556491	0.473528
	0.2	0.834000	0.7925	0.530862	0.456223
0.00005	**0.1**	0.743500	0.7435	0.666750	0.666750
0.00006	**0.0**	0.756333	0.7040	0.630707	0.544841
	0.2	0.802278	0.7465	0.587041	0.497051

Table 2: Results of the tuning for both models. Minimum and average Loss and Accuracy across all the tuning experiments performed.

The training was performed in a server with an Intel Dual Core processor and Nvidia Titan X GPUs, with approximately 32GB of memory. The run time varies a lot depending on the model. The average run time on the stage 1 dataset for model 1 is from 1 to 2 hours while for model 2 it has a run time of about 4 hours. For the training set for stage 2, the duration was 4 hours 37 minutes for model 1 and 8 hours 42 minutes for model 2. The final list of hyperparameters is in table 3.

[3] https://www.kaggle.com/c/gendered-pronoun-resolution/overview

Parameter	Value
Optimizer	Adam
Vocabulary Size	28996
Dropout	0.1
Sequence Length	256
Batch Size	32
Learning Rate	$3e-5$
Warm Up	20%
Steps	Stage 1: 81 — Stage 2: 148
Epochs	1
Gradient Accumulation Steps	5

Table 3: Hyperparameters for the model training

4 Results

Tables 4 and 5 report results for models 1 and 2 reported in section 2.1 for stage 1 of the competition. Both models 1 and 2 have similar overall results. Also both models show problems with the None class, model 2 specially. We believe this is because our model is based on guessing the correct name, so the guessing of none is not as well suited to it. Also, the training set contains much less of these examples, therefore making it even harder to train for them.

	Precision	Recall	F1	Support
A	0.83	0.87	0.85	874
B	0.88	0.88	0.88	925
None	0.64	0.52	0.57	201
Avg	**0.83**	**0.84**	**0.84**	**2000**

Table 4: Model 1 results for the testing stage 1.

	Precision	Recall	F1	Support
A	0.81	0.86	0.83	874
B	0.88	0.78	0.82	925
None	0.48	0.62	0.54	201
Avg	**0.81**	**0.80**	**0.80**	**2000**

Table 5: Model 2 results for the testing stage 1.

4.1 Advantages of the Masked LM Model

As well as the Masked LM, other BERT implementations were experimented with for the task. First, a text multi class classification model (figure 4) where the *[CLS]* tag is placed at the beginning of every sentence, the text is passed through a pretrained BERT and then the result from this label is passed through a feed forward neural network.

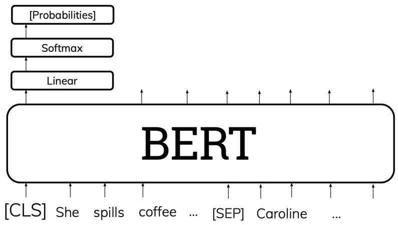


Figure 4: Model: BERT for text classification

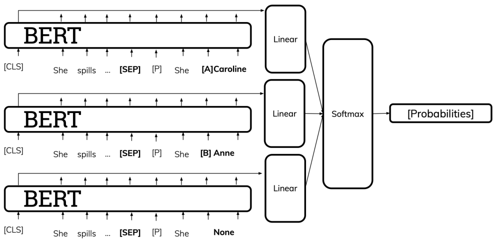


Figure 5: Model: BERT for multiple choice answering

And a multiple choice question answering model (figure 5), where the same text with the *[CLS]* label is passed through BERT with different answers and then the result these labels is passed through a feed forward neural network.

These two models, which were specifically designed for other tasks had similar accuracy to the masked LM but suffered greatly with the log loss, which was the competition's metric. This is because in a lot of examples the difference between the probabilities of one class and another was minimal. This made for a model where each choice had low confidence and therefore the loss increased considerably.

	Accuracy	Loss
BERT for Classification	0.8055	0.70488
BERT for Question Answering	0.785	0.6782
BERT for Masked LM	*0.838*	*0.44231*

Table 6: Results for the tests with different BERT implementations.

4.2 Name Replacement Results

As table 2.2 shows, name replacement considerably improved the model's results. This is in part because the names chosen as replacements are more common in BERT's training corpora. Also, a 43% of the names across the whole GAP dataset are made up of multiple words. So replacing these with a single name makes it easier for the model to identify their place in the text.

	Accuracy	Loss
Model 1 Original Names	0.782	0.7021
Model 1 Name Replacement	0.838	0.4423

Table 7: Results for the models with and without name replacement.

4.3 Competition results

In the official competition on Kaggle we placed 46th, with the second model having a loss around 0.301. As the results in table 8 show, the results of stage 2 were better than those of stage 1. And the second model, which had performed worse on the first stage was better in stage 2.

	Model 1	Model 2
Stage 1	0.44231	0.49607
Stage 2	0.31441	0.30151

Table 8: Results for both models across both stages of the competition

5 Conclusions

We have proved that pre-trained BERT is useful for co-reference resolution. Additionally, we have shown that our simple 'Name Replacement' technique was effective to reduce the impact of name frequency or popularity in the final decision.

The main limitation of our technique is that it requires knowing the gender from the names and so it only makes sense for entities which have a defined gender. Our proposed model had great results when predicting the correct name but had trouble with with the *none* option.

As a future improvement it's important to analyze the characteristics of these examples where none of the names are correct and how the model could be trained better to identify them, specially because they are fewer in the dataset. Further improvements could be made in terms of fine-tuning the weights in the actual BERT model.

Acknowledgements

This work is also supported in part by the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund and the Agencia Estatal de Investigación, through the postdoctoral senior grant Ramón y Cajal, contract TEC2015-69266-P (MINECO/FEDER,EU) and contract PCIN-2017-079 (AEI/MINECO).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sameer S. Pradhan and Nianwen Xue. 2009. OntoNotes: The 90% solution. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*, pages 11–12, Boulder, Colorado. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.