Ukrainian Catholic University

Faculty of Applied Sciences

Data Science Master Programme

Gendered pronoun resolution

Machine Learning project report

Authors:
Dmytro Babenko
Maksym Opirskyi
(contributed equally, order chosen alphabetically)

April 23, 2019

Abstract

We are going to investigate several possible solutions for gendered pronoun resolution problem. One of them is based on simple machine learning approach using One-vs-the-rest (OvR) multiclass strategy. Furthermore, we are going to solve this problem using more sophisticated natural language processing algorithms which adopt deep learning.

Keywords: Pronoun resolution · One-vs-the-rest (OvR), BERT (Bidirectional Encoder Representations from Transformer), MLP (Multilayer perceptron), multiclass strategy, Conv1d · NLP (Natural Language Processing)

1 Introduction

Coreference resolution is a well-studied problem in the field of natural language processing. A subtask of the coreference resolution which is this report dedicated to has however one major flaw. Recent studies show that there is a gender bias among state-of-the-art coreference resolvers. Google AI Language recently released Gendered Ambiguous Pronouns (GAP) dataset [1], containing gender-balanced pronouns, meaning that this dataset contains 50% of feminine pronouns and 50% of masculine. The Kaggle platform recently started a challenge that uses this dataset. The aim of the challenge is to build the best gendered pronoun resolution model.

In this report, we are going to implement and compare several possible solutions to the mentioned problem based on machine learning (OvR multiclass strategy) and deep learning algorithms.

2 Importance of the problem

2.1 Motivation

Pronoun resolution is a part of coreference resolution, the task of pairing an expression to its referring entity. This is an important task for natural language understanding, and the resolution of ambiguous pronouns is a longstanding challenge. Obtaining effective coreference resolution algorithms would give a huge boost to other NLP fields such as machine translation, sentiment analysis, paraphrase detection, summarization, etc.

2.2 Problem formulation

Suppose there is a set of samples each of which consists of one or several sentences where two different subjects are considered to be candidates for a reference. There is also a target pronoun which corresponds to one of these two subjects.

Figure 1: The example of pronoun problem.

As we can see from Figure 1, we have a text and it has two persons *Jose de Venecia Jr.* and *Abalos*. Additionally, the text has pronoun *him* and we should define to which of the two candidate subjects is this pronoun related. Each sample is labeled, that is we know the true reference. This task can therefore be viewed as a supervised classification problem. In fact, it is multiclass, since there is a fraction of instances where neither of the candidate subjects corresponds to the target pronoun.

3 Data

3.1 Collected data

As this project is a part of Kaggle competitions, there is available labeled training set on GAP Dataset Github Repo. This link was provided by Kaggle, because unlike many Kaggle challenges, this competition does not provide an explicit labeled training set.

	ID	Text	Pronoun	Pronoun-offset	A	A-offset	A-coref	B	B-offset	B-coref	URL	NEITHER
0	test-1	Upon their acceptance into the Kontinental Hoc...	His	383	Bob Suter	352	0	Dehner	366	1	http://en.wikipedia.org/wiki/Jeremy_Dehner	0.0
1	test-2	Between the years 1979-1981, River won four lo...	him	430	Alonso	353	1	Alfredo Di St*fano	390	0	http://en.wikipedia.org/wiki/Norberto_Alonso	0.0
2	test-3	Though his emigration from the country has aff...	He	312	Ali Aladhadh	256	1	Saddam	295	0	http://en.wikipedia.org/wiki/Aladhadh	0.0
3	test-4	At the trial, Pisciotta said: ``Those who have...	his	526	Alliata	377	0	Pisciotta	536	1	http://en.wikipedia.org/wiki/Gaspare_Pisciotta	0.0
4	test-5	It is about a pair of United States Navy shore...	his	406	Eddie	421	1	Rock Reilly	559	0	http://en.wikipedia.org/wiki/Chasers	0.0

Figure 2: Available datasets.

From Figure 2 above, we can see that this set has 11 columns: ID, Text, Pronoun, Pronoun-offset, A, A-offset, A-coref, B, B-offset, B-coref and URL.

- ID - id of sample item
- Text - the text itself (one or couple sentences).
- Pronoun - the pronoun in the text which should be related to A or B subject
- Pronoun-offset - start position of the pronoun in the text
- A - first subject in the text
- A-offset - start position of the subject A in the text
- A-coref - boolean value which shows whether considered pronoun relates to subject A
- B - second subject in the text
- B-offset - start position of the subject B in the text
- B-coref - boolean value which shows whether considered pronoun relates to subject B
- URL - link to the site where this text was found

3.2 Preprocessed data

It is clear that these features are not enough to train our model. So we constructed new features based on the available ones:

- Pronoun-offset2 - end position of pronoun in the text
- A-offset2 - end position of subject A in the text
- B-offset2 - end position of subject B in the text
- A-dis - distance between subject A and pronoun (in characters)
- B-dis - distance between subject B and pronoun (in characters)
- section_min - minimum of Pronoun-offset, A-offset, B-offset
- section_max - maximum of Pronoun-offset, A-offset, B-offset

The example of extended dataset can be viewed below on the Figure 3.

	ID	Text	Pronoun	Pronoun-offset	A	A-offset	A-coref	B	B-offset	B-coref	URL	NEITHER	Pronoun-offset2	A-offset2	B-offset2	A-dist	B-dist	section_min	section_max
0	test-1	Upon their acceptance into the Kontinental Hoc...	His	383	Bob Suter	352	0	Dehner	366	1	http://en.wikipedia.org/wiki/Jeremy_Dehner	0.0	386	361	372	31	17	352	386
1	test-2	Between the years 1979-1981, River won four lo...	him	430	Alonso	353	1	Alfredo Di St*fano	390	0	http://en.wikipedia.org/wiki/Norberto_Alonso	0.0	433	359	408	77	40	353	433
2	test-3	Though his emigration from the country has aff...	He	312	Ali Aladhadh	256	1	Saddam	295	0	http://en.wikipedia.org/wiki/Aladhadh	0.0	314	268	301	56	17	256	314
3	test-4	At the trial, Pisciotta said: ``Those who have...	his	526	Alliata	377	0	Pisciotta	536	1	http://en.wikipedia.org/wiki/Gaspare_Pisciotta	0.0	529	384	545	149	10	377	545
4	test-5	It is about a pair of United States Navy shore...	his	406	Eddie	421	1	Rock Reilly	559	0	http://en.wikipedia.org/wiki/Chasers	0.0	409	426	570	15	153	406	570

Figure 3: The example of additional linguistic features.

Obviously, using these features did not improve accuracy significantly, so we focused on adding more. A great amount of information could be gained adding linguistic features.

3.3 Linguistic features

There were extracted additional linguistic features using spacy library and BERT network.

Initially, we acquire additional features using spacy package. We created the Stanford Dependencies (SD) representation [7] for text (one or couple sentences). After that, we know what each subject in the text has Penn Treebank tags [8]. We found top five tags which occur the most often for subject A and B. These are the following tags:

- conj - conjunct
- dobj - direct object
- poss - possession modifier
- pobj - object of a preposition
- nsubj - nominal subject

Then, additional 10 columns which show how many times specific tag occur for specific subject were created. See example of these additional features on Figure 4.

A-poss	B-poss	A-nsubj	B-nsubj	A-pobj	B-pobj	A-dobj	B-dobj	A-conj	B-conj
0	1	0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0

Figure 4: The example of linguistic features.

Furthermore, we extracted more linguistic features using BERT embeddings. We used three vectors, which relate to subject one, subject two and pronoun.

4 Modeling

4.1 Baseline model

To start with, we tried a classical machine learning approach to solve a problem. Since we deal with supervised multiclass classification problem, the common thing is, having the text, to construct features that are able to grasp references between candidates and target pronoun as well as possible. These are typically combination of distance features (e. g. distance between pronoun and candidate, in characters or in words) and linguistic features like syntactic dependencies, gender or number agreement. The features are fed into classifier, the most common is decision tree [2] [3] [6]. We experimented with various classifiers like decision trees, logistic regression, random forest and XGBoost. They all showed similar and quite unpleasant result. Random forest classifier turned out to be the best with a very little difference to other classifiers.

4.2 BERT embeddings

The approach described above is often not sufficiently effective, since it depends on manually constructed features which makes it unable to generalize well and hard to maintain. Moreover, the text possesses much greater amount of information which can be used to learn relations between words. With rise of deep learning words could be represented as vectors that carry information about semantic dependencies [4]. Each word is embedded in a vector space such that vectors representing words that have similar semantic meaning are close in this space. This allows to obtain more useful information from the given data, comparing to the limited knowledge that give handcrafted features. Furthermore, such an approach significantly reduces the need to create features manually, which can be a time consuming end error prone process.

Having the vector representations of words in each training sample one of the possible solutions to coreference problem would be to feed embeddings of candidate subjects and a pronoun into a classifier. To create word embeddings we used BERT which is a state -of-the-art model for general-purpose "language understanding". BERT is capable of extracting contextual representations in a deeply bidirectional manner - meaning that it considers both left-side and right-side contexts of a word [5].

4.3 Neural models

Apart from simple classifiers that did not get a satisfactory result we used multilayer perceptron with one hidden layer. As input we simply use concatenated word embeddings corresponding to pronoun and to candidate subjects and handcrafted features created before. Since with the use of BERT word representations the dimensionality of the input space grew significantly we decided that many machine learning algorithms will not suffice in this case. MLP on the other hand can handle such an amount of features.

Figure 5: MLP Classifier.

We also had the idea that due to the ordered nature of the embeddings (permuting the entries in a word vector would eliminate all its usefulness) convolution layers would help to increase the accuracy of the prediction. So we used one dimensional convolutional neural network.

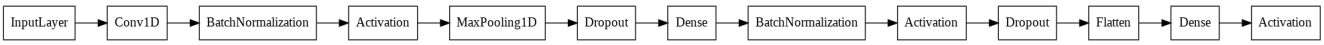


Figure 6: Conv1d Classifier.

We experimented with both model tweaking hyperparameters like layer sizes, layer number, filter number for convolutional layer, dropout rate.

5 Evaluation

The challenge required evaluation metric for this problem to be multi-class logarithmic loss. For each training sample we output predicted probabilities of target pronoun to reference one of the subjects or neither of them. The function to minimize is then:

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(p_{ij})$$

where N is the number of samples, M is number of classes and is therefore equal to 3 (A, B or NEITHER), y_{ij} equals 1 when sample i belongs to class j and p_{ij} is predicted probability that sample i belongs to class j.

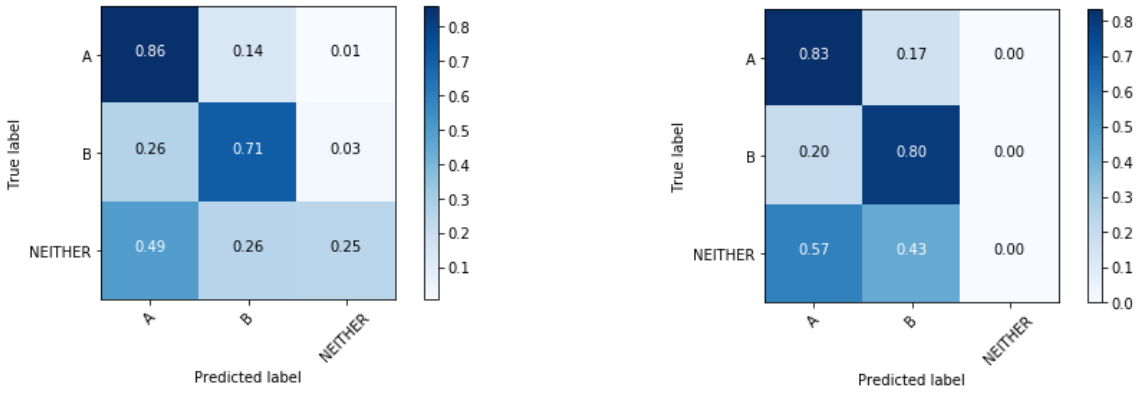
5.1 One-vs-the-rest multiclass strategy

Initially, we trained our model using One-vs-the-rest (OvR) multiclass strategy. In this case, we tried training with different feature matrices and different classification algorithms. Each time, this feature matrix was extended.

Firstly, we trained this model with feature matrices described on Figure 2 and Figure 3 and got log loss 0.9188 and 0.9127 respectively. As described above, these feature matrices contained symbol offsets and symbol distances parameters.

Secondly, we trained the same model with feature matrix extended by linguistic features, described in section 3.3. Here, we got better log loss - 0.7692.

Thirdly, we added another offset and distance features. Compared to initial offsets and distances, these parameters were calculated as word offset (position of the word in the text) and word distance (the number of words between subject and pronoun). Furthermore, we added two more features: similarity between vector of subject



(a) Conv1d Network

(b) MLP

Figure 7: Confusion matrices for embeddings only input

one and pronoun vector, similarity between vector of subject two and pronoun vector. Vectors for these words were built using BERT embedding approach. It helps us to build the vectors of the words in the specific text, after that we can easily calculate cosine similarity between these vectors. The last log loss, which we got with finally extended feature matrix was a little bit better than previous, that is 0.7051.

5.2 Neural models

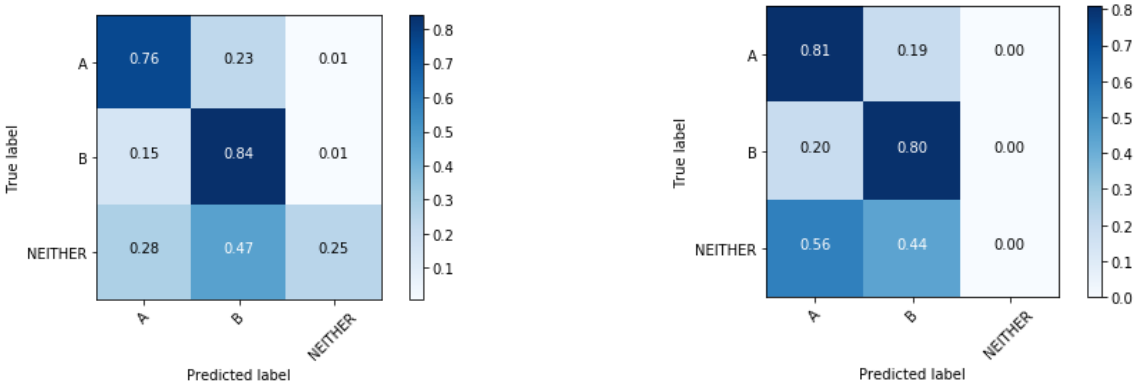
Firstly, we trained the models using only BERT embeddings as input. In this case MLP showed loss of 0.6666 and Conv1d Network had 0.6456. Figure 7 demonstrates confusion matrices for both models.

Next, we trained the models with all features we have - that is embeddings along with manually constructed features. In this case we got log loss 0.6508 for MLP Classifier and 0.6142 for Conv1d Network. Figure 8 shows confusion matrices for this type of input. We observe that both models heavily misclassify third class - when pronoun corresponds to neither of the subjects. Matrices are normalized, but there are little "Neither" samples in the set. Interesting is the fact that MLP predicts no sample to be "Neither", while Conv1d Network does a better job.

Although additional features turned to be useful for both models, the improvement was not significant. However, in case of Conv1d Network the loss decreased more. All in all, Conv1d Network showed better result.

6 Conclusions

To summarize, we tried to solve the problem from Kaggle competition with different methods. We trained various models (OVR classifiers, MLP, Conv1d Network) based on available datasets. Besides, we extended features which gave us better results. In addition, we used BERT embeddings, and the log loss improved significantly. OVR classifiers did not showed satisfactory result, so we decided to use neural networks for classification. We created two models - MLP and Conv1d Network and experimented with inputs for them. These models showed better result, particularly because of BERT since it is state-of-the-art model for creating word representations that considers semantic relationships between words in the text. At this stage we stopped. However, there is a lot of space for improvement. One of the possible extensions to our solution would be to consider more embeddings as input - not only those corresponding to pronoun and candidates, but also embeddings of the words located near the initial ones. We leave this as a future work. Experiments done so far are available on Kaggle public kernels [9] and [10].



(a) Conv1d Network

(b) MLP

Figure 8: Confusion matrices for all features input

References

[1] Webster, K., Recasens, M., Axelrod, V., Baldridge, J: Mind the GAP: A Balanced Corpus of Gendered Ambiguous Pronouns. CoRR, abs/1810.05201 (2018)

[2] Soon W.-M, Ng, H.-T., Lim, D.-C.-Y.: A machine learning approach to coreference resolution of noun phrases. Comput. Linguist. 27, 4 (December 2001), 521-544 (2001).

[3] Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02). Association for Computational Linguistics, Stroudsburg, PA, USA, 104-111 (2002). DOI: https://doi.org/10.3115/1073083.1073102

[4] Sukthanker, R., Poria, S., Cambria, E., Thirunavukarasu, R.: Anaphora and Coreference Resolution: A Review

[5] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805

[6] Mccarthy, J., Lehnert, W.: Using Decision Trees for Coreference Resolution.

[7] De Marneffe, M.-C., Manning, C. D.: Stanford typed dependencies manual. Technical report, Stanford University, 2008.

[8] https://www.sketchengine.eu/modified-penn-treebank-tagset/

[9] https://www.kaggle.com/dmytrobabenko/simple-ml-model

[10] https://www.kaggle.com/needdatasets/simple-ml-model.