NLP course project. Author gender prediction

Fedor Makarov, Rufina Galieva

INTRODUCTION

In this project we are going to examine a part of author profiling task (characterization of an author through some key attributes), particularly focusing on gender prediction given a text written by a person. The task is challenging because men and women use same words, can discuss same topics in their tweets, hence making the task insoluble. The task becomes even harder for English language since it does not have any identifiers for male and female as Russian language for instance.

RELATED WORKS

Author gender prediction was already touched by researchers in machine learning. For example, people already tried to aplly such methods as **Convolutional Neural Networks** in the work of Sezerer et al. (2018) [5]. But this work makes a gender prediction based on multiple tweets of a user. Another approach is **Support Vector Machines** used by Khandelwal et al. (2018) [6], examining different kind of methods to extract features from text (Character N-grams, Bag of words, Reference tokens). This work has its own collected dataset as well. Another interesting work applied **Recurrent Neural Networks** for the task by Sezerer et al. (2019) [4]. And again, this work uses multiple user tweets to identify the gender.

DATA SET

In this project we use PAN 2018 Author Profiling data set [1] for training and evaluating the model. Data set contains 413,555 different tweets, 209,544 of which correspond to male and 204,011 correspond to female. Corpus contains 725,800 unique words. Shortest tweet in the corpus contains 1 sentence, longest tweet contains 5,421 sentences. 99% of tweets contain less than or equal to 100 sentences. Shortest sentence contains 1 word, longest sentence contains 32,123 words. 99% of the data contain less than or equal to 400 words. According to the dataset most of the time women use around 500 words in the tweet, while men use around 15.

METHODOLOGY

Data preprocessing

Coming from Twitter data contains a lot of "trash", such as html tags, one letter words, misspelled words. Preprocessing tweets before feeding into model is an essential part of the pipeline. Therefore we first remove html from the data, then tweets are converted to lower case, then tweets are tokenized, words that occur less than 3 times are removed. After that we obtain 178,824 unique words.

During processing tweets we obtain that men and women use most of the time use same words. The only difference between men and women that we obtained is the length of the tweet, in general men write short tweets and women write long tweets, however this fact can not be generalized to all tweets.

Models

We explore different kinds of tweets representations as well as different models applied to that representations. Each method is described below.

Manual features

We manually selected 7 features: the length of the tweet, the average length of the word, the use of numbers on the tweet and the percentage of the main parts of speech.

MLP model with one hidden layer of 1024 elements showed an accuracy of 56%. An analysis of the effect of features on the result was carried out and presented in figure 1. As we can see first feature is very strong - the length of the tweet. Second significant features - average word length and percentage of adverbs. The percentage of nouns and adjectives is not significant, verbs and numbers in general are essentially equal to the statistical error.

Weight	Feature
0.0225 ± 0.0020	Length of tweet
0.0123 ± 0.0043	Average length of words
0.0094 ± 0.0013	Pecentage of ADVs
0.0064 ± 0.0028	Percentage of NOUNs
0.0052 ± 0.0025	Pecentage of ADJs
0.0034 ± 0.0027	Pecentage of VERBs
0.0024 ± 0.0028	Pecentage of numbers used

Figure 1. Manully selected feature importances.

Link to implementation in colab: https://colab.research.google.com/drive/1gS5nRRJu2phAJp2VerttJ0295wtW5cSk

TF-IDF + MLP + bag of words

There are more than 300.000 unique words in the dataset. Size of a vector is too large to process it. But most of words occurs very rare. We cut the tale at the level of 100 words, which occurs in a dataset and after it we have 20031 unique words. The model is MLP, with 2 hidden layers with size of 256 and 32 elements. It still couldn't be processed without using batches, but at least it fitted into colab RAM. Also we added a dropout because loss has gone too far from validation loss. In fact, simplicity of a model and size of a dataset allowes us to get a good result even after 1st epoch. So we can estimate final accuracy of this model about 57%. Training progress of TF-IDF + MLP + bag of words model is illustrated in figure 2.

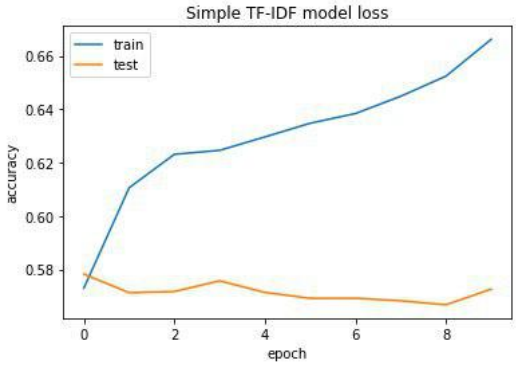

Figure 2. TF-IDF + MLP + bag of words training progress.

Link to implementation in colab: https://colab.research.google.com/drive/1gS5nRRJu2phAJp2VerttJ0295wtW5cSk

Bag-of-words representation

One of approaches in this work is one of simplest representations of documents - Bag-of-words. Since our data contains 178,824 unique words each tweet will be represented with vector of length 178,824. We lose word order information keeping only information on word occurrences. Hence this representation did not give good results together with multi-layer perceptron (MLP). We applied only MLP together with bag-of-words because of the dimensionality, LSTM work long in that case, classical machine learning struggle here as well.

Link to implementation in colab: https://colab.research.google.com/drive/1rh4JxqEN5oDb08bt5S0CXNRYOr7vQ8om

Fasttext word embeddings

Another simple approach to try is FastText word embeddings pretrained on *yahoo answers* dataset. We obtain tweet representation by averaging word embeddings of words in the tweet. Then we feed that tweets representations to Logistic Regression (LR), Linear Discriminant Analysis(LDA), Multilayer Perceptron (MLP) and LSTM.

We did same experiments with **Glove** pretrained on twitter data, however could not obtain any improvements.

Link to implementation in colab: https://colab.research.google.com/drive/1rh4JxqEN5oDb08bt5S0CXNRYOr7vQ8om

Trainable word embeddings

Just as in case of FastText we learn embedding vectors to words in the tweet, then average them and feed to classification layer of the model. We limit the length of the tweet to 5000 words and apply zero padding to tweets shorter than that. Overall pipeline of model with trainable embeddings is illustrated in figure 3.

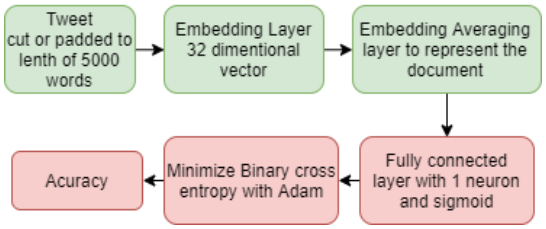

Figure 3. The trainable embeddings model pipeline.

Link to implementation in colab: https://colab.research.google.com/drive/10UTs9jfzuhSTm5YJvsdBFpnJ3qvpapc4

CNN

Convolutional Neural Networks are usually applied to images, but instead of image pixels, the input to most NLP tasks are sentences or documents that can be represented as a matrix. Each row of the matrix corresponds to one token, typically a word. That is, each row is vector that represents a word.

In this work we limit tweets to the length of 5,000, if tweet contains more than 5,000 words we cut the tweet, if tweet contains less than 5,000 we apply post zero padding to the tweet.

We first feed tweets to the embedding layer of the model obtaining matrix of size 5,000 x 32. Then three convolution operations are applied separately for the matrix. Then only highest features are selected using max pooling operation and resulting features are concatenated and fed to the fully connected layer. We also explored using average pooling, but results was a little worse.

The proposed CNN model architecture is illustrated in figure 4.

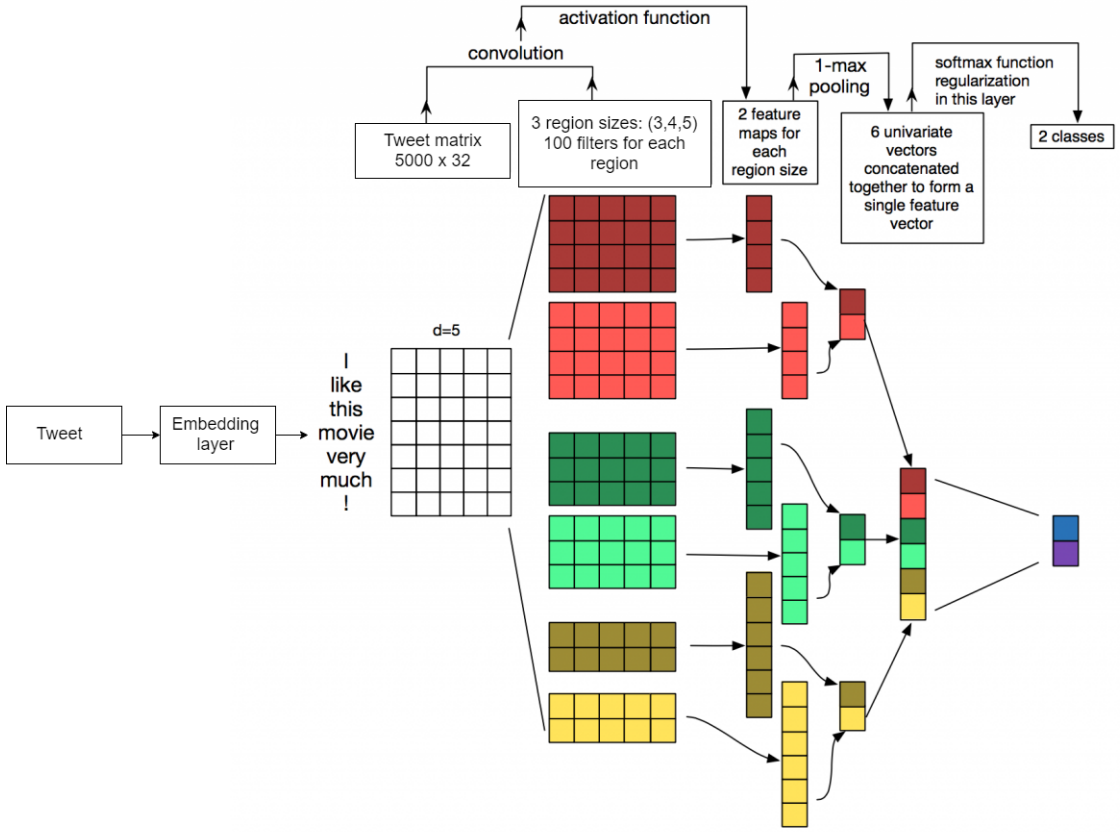

Figure 4. The proposed CNN model.

Link to implementation in colab: https://colab.research.google.com/drive/1DY9ecKQdEvYAwbLjyJpATHVhtmsUF7RG

EVALUATION METRICS

In order to evaluate our model the standard metric of accuracy is used as a performance measure. It is computed as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \qquad \textbf{Equation } 1.$$

Where, TP is the number of true positives, TN is the number of true negative, FP is the number of false positives and FN is the number of false negatives.

RESULTS

In this section we report results of all models that we have tried. Comparison of differnt models based on evaluation metrics is presented in table 1.

Model	Accuracy
Manual features	56%
TF-IDF + MLP + bag of words	57%
Bag-of-words + MLP	57%
FastText + LR	52%
FastText + LDA	54%
FastText + LSTM	57%
FastText + MLP	56%
Trainable embeddings	60%
CNN	**62,7%**

Table 1. Results of models evaluation on test set of PAN 2018 data.

CONCLUSION

In this work we applied many different models as well as tweet representations, but all of them give almost the same results, accuracy around 55%, which means that most of the words contained in the tweets are used by both female and male in relatively equal proportions, at the same time there are some features, topics that distinguish male and women, but they do not occur in all tweets of the person, hence we can distinguish only in 50% of cases. People have tweets on has neutral topics and gender specific topics, for example, for women it's pregnancy, cosmetics, at the same time the themes of war, some specific computer games, are almost certainly men. If we take a bunch of tweets for a given person we could most likely identify their gender, as done in previous works, because among all tweets there will be neutral topics as well as gender specific topics.

REFERENCES

[1] Francisco Manuel Rangel Pardo, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches. Overview of the Author Profiling Task at PAN 2013. In Pamela Forner, Roberto Navigli, and Dan Tufis, editors, CLEF 2013 Evaluation Labs and Workshop - Working Notes Papers, 23-26 September, Valencia, Spain, September 2013. CEUR-WS.org. ISBN 978-88-904810-3-1. ISSN 2038-4963. *https://pan.webis.de/data.html*

[2] J. Schler, M. Koppel, S. Argamon and J. Pennebaker (2006). Effects of Age and Gender on Blogging in Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs. *http://u.cs.biu.ac.il/ koppel/BlogCorpus.htm*

[3] *https://bitbucket.org/lowlands/release/src/fd60e8b4fbb12f0175e0f26153e289bbe2bfd71c/WWW2015/data/?at=master*

[4] Sezerer Erhan, Ozan Polatbilek, and Selma Tekir. Gender Prediction from Tweets: Improving Neural Representations with Hand-Crafted Features (2019). *https://arxiv.org/pdf/1908.09919.pdf*

[5] Erhan Sezerer, Ozan Polatbilek, Özge Sevgili, and Selma Tekir. Gender Prediction From Tweets With Convolutional Neural Networks (2018). *http://ceur-ws.org/Vol-2125/paper*$_1$16*.pdf*

[6] Khandelwal, Ankush, et al. Gender Prediction in English-Hindi Code-Mixed Social Media Content: Corpus and Baseline System. (2018) *https://arxiv.org/pdf/1806.05600.pdf*

[7] FastText. Library for efficient text classification and representation learning. *https://fasttext.cc/*

[8] GloVe: Global Vectors for Word Representation. *https://nlp.stanford.edu/projects/glove/*

[9] Mikolov, Tomas, et al. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781 (2013).*