Rob's Angels: Embedding and Clustering for Cross-Genre Gender Prediction

Rianne Bos d.bos.7@*
Kelly Dekker k.dekker.5@*
Harm-Jan Setz h.setz@*

*student.rug.nl

Abstract

For CLIN 2019 a shared task on binary gender prediction within and across different genres in Dutch was issued. This paper reports on the findings of team 'Rob's Angels' done in light of this shared task. A multitude of linear SVM models were created to predict gender in different genres (Twitter, YouTube and news), and cross-genre. Our best models used Twitter word-embeddings, in combination with removal of stopwords and tokenization of the text. We also introduced a novelty in classifying the news corpus. The large instances of news data are split into smaller parts, individually classified, and then the text as a whole is assigned a label based on majority voting. We eventually finished eighth on the in-genre category with an average accuracy of 0.617 and fourth on the cross-genre category with an average accuracy of 0.547.

1 Introduction

Picture a news article. It starts with a title, then a short intro, followed by the name of the author. In many cases (if the name is not ambiguous) you do now know if the author is a man or a woman. In case of a tweet, you often can determine gender from the profile picture. But what if you remove this kind of information and are left with only the actual text? If the text is written on paper, you could try to infer gender from handwriting, as Hamid and Loewenthal (1996) did. But nowadays, many texts are written in an electronic environment. Does this mean that there is no way of telling if a text is written by a man or a woman? This is the task of gender identification. Can you, based on the text someone has written, predict if this person is either a man or a woman? In this paper, we examine three kinds of text, from different genres: tweets, YouTube comments and news articles. Both, in-genre and cross-genre predictions are done, i.e. train on one genre and predict for the same genre and train on everything but one genre and test on that left out genre. The main focus will be on the cross-genre task, since the "official" winner of the shared task is determined based on the cross-genre ranking. The best cross-genre model will also be used for the in-genre predictions.

2 Background

Gender prediction is something which is a subtask of author profiling. Much previous work has been done on author profiling, but most of it was conducted on a single domain. However the focus of this shared task is on cross-genre modelling. A similar shared task has been held for Italian texts (Dell'Orletta and Nissim, 2018). The participants were provided with different genres: Twitter, YouTube, Children writing, Journalism and Personal diaries. Given texts from these genres, the task was to predict gender in-domain and cross-domain. The datasets for both training and test were balanced in terms of gender label distribution (50:50). For in-genre prediction, the highest score was obtained for the personal diaries genre, namely an accuracy of 0.676. In a cross-genre setting, the highest accuracy was 0.640 for the Children writing genre. The hardest genre seemed to be Journalism, since for both in- and cross-genre this genre had the lowest average scores when taking all teams into account. Multiple models have been used, like an SVM, Logistic Regression, Random Forest or Bi-LSTM. Overall, neural models performs slightly better than a classis SVM, but the difference is minimal.

Another similar shared task has been done on

Russian as well (Litvinova et al., 2017). The set up however is slightly different since participants were only given a training set on Twitter. Their models were evaluated on five genres: essays, Facebook, Twitter, reviews and text where the authors imitated the other gender. Again, multiple machine learning techniques have been used like SVM's or neural networks. Deep learning techniques seemed to work best for the essay and gender imitation genre, while a SVM with combinations of n-grams performed better on the Facebook, Twitter and reviews genre.

So, judging from the results of these similar shared tasks, both SVM's and neural networks seemed to perform well for different genres. Due to the simplicity and efficiency of SVM as a machine learning technique, we will mainly focus on SVM models.

3 Data

The data used for this shared task is in Dutch and is retrieved from the following three genres: Twitter, YouTube and news. For Twitter, a set of tweets is provided, for YouTube a set of video comments and for news a set of articles. Every instance of text in the dataset had an id number, a genre label and a gender label. See below for a sample instance from the Twitter data:

```
<doc id="36" genre="twitter" gender="F">
Hou van mijn vrienden
</doc>
```

Among all genres, the gender distribution was balanced (50:50). Table 1 shows the distribution of data between the three genres.

The number of instances differ a lot between the three genres, but the number of tokens are of more comparable sizes. The amount of instances of the news genre is relatively low when compared to the other two genres, but the number of average tokens per instance reveal that these texts are considerably bigger.

	Twitter	YouTube	News
Instances (I)	20,000	14,744	1,832
Tokens (T)	469,105	300,360	382,146
Avg. T/I	23.5	20.4	208.6

Table 1:
Data distribution of the genres

4 Method

The main focus of this paper is on the cross genre task. For this task we first decided to train on Twitter data to predict on the YouTube data, train on the YouTube data to predict on the Twitter data and train on the Twitter and YouTube data together, to predict on the news data. This is because the Twitter and YouTube data have more or less the same length per message and the messages contain less standard text. This will probably lead to the use of different kind of words than in the news texts. The decision to use both Twitter and YouTube to train for the prediction of news is based on the fact that we wanted as much training data as possible. This is because the style of the news genre is completely different from the Twitter and YouTube.

Preprocessing

We experimented with a variety of preprocessing steps:

- removal/replacement of emojis: use regular expressions to find emoticons, symbols & pictographs, transport & map symbols and flags.
- removal/replacement of username: use regular expressions to find usernames (starting with @)
- removal of stopwords: While using embeddings we removed stopwords with the NLTK stopwords-list, to get a more accurate average embedding.
- tokenization: use nltk word tokenizer to separate the words
- POS-tagging: use POS-tags from spacy model nl_core_news_sm
- removal of links: use regular expressions to find the links (starting with http(s))

In case of replacement, emojis were replaced with '<EMOJI>' and usernames were replaced with '<USERNAME>'.

Classifier

We experimented with a support vector machine (SVM), specifically a LinearSVC. For the tokenized words and the POS-tags we experimented with a Tfidf-vectorizer with different n-gram ranges (unigrams, bigrams and trigrams and combinations of these 3).

Split news

In order to make the news data more similar to the short Twitter and YouTube texts, we split up the news articles. We experimented with a split every two, three and four sentences. Only the last split could consists of less sentences, because e.g when you split every three sentences and a article has 20 lines, the last split will consist of 2 sentences in stead of 3. For every split the system predicts if the gender is male or female, and in the end majority voting is used to get the final gender. In case of an equal score, the gender is set to female. This is purely based on that female gave higher scores than male for the given data.

Brown clusters

We also experimented with the use of Brown clusters. The Brown clusters that are used are created by Bouma (2015), on the basis of 594 million Dutch tweets (5.8 billion tokens). They are collected between 2011 and 2014, using the method described in the paper of Sang and van den Bosch (2013). We used the Brown clusters to replace all tokens with the corresponding cluster. When the token did not have a corresponding cluster, we assigned it the value 'unk'. For each sentence, the clusters where then put together in one string.

Word embeddings

Our best model used Twitter embeddings created as described in the paper by van der Goot and van Noord (2017). However, for these embeddings the default settings of the word2vec are used, but with skip-grams. We translated each token of each instance into it's 100 dimensional embedding. When a token had no equivalent embedding, we skipped this token. When we had a list of 100 dimensional embeddings for each instance of the data-set, we then used the average 100 dimensional embedding of each instance, and fed this to the classifier as features.

5 Results

The baseline for this shared task was a majority baseline of an accuracy of 50%. Our best results on the development and test set on in-genre classification are reported in Table 2. The classifier used to get these results is a LinearSVC with the C parameter set to 1. The results of the different features and pre-processing steps for cross-genre classification on the development data are reported in Table 3, 4 and 5.

Genre	Development	Test
Twitter	0.648	0.648
YouTube	0.604	0.594
News	0.633	0.609
Avg.	0.628	0.617

Table 2:
In-genre results on development and test data

Features	VecPOS	VecTOK	Trained on	Acc.
F1	X	Tfidf()	Twitter	0.535
F1, F2	Tfidf()	Tfidf(n_range = 1,3)	Twitter	0.524
F1, F2, F3	Tfidf()	Tfidf(n_range = 1,3)	Twitter	0.549
F1, F4	X	Tfidf()	Twitter	0.536
F1, F3, F5, F6	X	X	Twitter	**0.570**

Table 3:
Cross-genre results for YouTube[1]

Features	VecPOS	VecTOK	Trained on	Acc.
F1	X	Tfidf()	YouTube	0.527
F1, F2	Tfidf()	Tfidf(n_range= 1,3)	YouTube	0.552
F1, F4	X	Tfidf()	YouTube	0.549
F1, F3, F5, F6	X	X	YouTube	0.570
F1, F3, F5, F6	X	X	News	**0.584**

Table 4:
Cross-genre results for Twitter[1]

Features	VecPOS	VecTOK	Trained on	Acc.
F1	X	Tfidf()	T + Y	0.545
F1, F2	Tfidf()	Tfidf(n_range= 1,3)	T + Y	0.541
F1, F4	X	Tfidf(n_range= 1,2)	T + Y	0.530
F1, F2, F4	Tfidf()	Tfidf(n_range= 1,3)	T + Y	0.552
F1, F4, F7	X	Tfidf(n_range= 1,2)	T + Y	0.541
F1, F3, F7	X	Tfidf(n_range= 1,3)	T + Y	0.550
F3, F7	X	X	Twitter	0.554
F3, F5, F7	X	X	Twitter	0.548
F4, F7	X	Tfidf()	Twitter	0.540
F1, F3, F5, F6	X	X	Twitter	0.553
F3, F5, F6, F7	X	X	Twitter	**0.557**

Table 5:
Cross-genre results for news[1]

All our best models, both cross- and in-genre, used the following preprocessing steps: tokenization, link removal, stopwords removal, replacement of usernames and the replacement of emojis. Furthermore, all systems used a 100 dimension word embedding model which was trained on Twitter data. The model used unigrams only,

[1]F1 = tokenization, F2 = POS-tagging, F3 = replace username + emojis, remove links, F4 = Brown clusters, F5 = word embeddings, F6 = remove stopwords

since the embedding model was trained on unigrams only.

For in-genre classification, the preprocessing step stopwords removal, was not incorporated for the YouTube model. The YouTube model performed less on the development data when stopwords were removed, while the Twitter and news models performed better.

The use of Brown clusters improved the accuracy scores on Twitter and YouTube slightly compared to using n-grams, but on news accuracy scores dropped. When we started working with the Twitter embeddings, we found out that training on news and testing on Twitter gave better results, compared to training on YouTube. For testing on YouTube, training on news with embeddings did not improve the results, probably because the embeddings are trained on Twitter data. We also discovered that, using embeddings, training on only Twitter data to predict on news improved the results over training on Twitter and YouTube together.

Only the news model used a slightly different approach. As described in Section 4, news articles were split up every three sentences and classified according to majority voting. In case of a tie, the female label was given since this resulted in higher scores for the development data.

The results on the test set on cross-genre classification are shown in Table 6.

Genre	Trained on	Test
Twitter	News	0.555
YouTube	Twitter	0.559
News	Twitter	0.528
Avg.		0.547

Table 6:
Cross-genre results on test data

6 Discussion

The main focus of our research was to find improvements for cross-genre gender prediction models. After initial low results with using word n-grams and POS-tags, experimented with Twitter-embeddings. Ideally we would have gotten embeddings trained on YouTube and news data as well, but we did not have another dataset for these genres, and were limited by time. We considered training on the test corpus of these genres, but they were too small for training embeddings. Also, training embeddings on a corpus that we had to test on seemed like cheating.

The implementation of the Twitter embeddings boosted our accuracy scores. Unsurprisingly the cross-genre predictions on the Twitter data got the biggest boost, but the other two genres also saw an increase in accuracy scores. The most surprising finding in our results was the high accuracy score we got on the Twitter corpus, with a model trained on news data. At first we did not even try this option, since the Twitter and news corpus are fundamentally different, and we thought we would get better results training on the YouTube data. However, when training on the YouTube corpus we got an accuracy of 0.57 on Twitter, while when training on news we got 0.584.

Since the instances in the news corpus have on average much more tokens than instances in the other two corpora, our results would suggest that for cross-genre training, it is better to get an average embedding for each label over a larger amount of data. This extra data seems to increase the quality of the embedding.

7 Conclusion

Based on our results, we can conclude that using embeddings as features for an SVM-classifier works better in cross-genre gender prediction than using Brown clusters as features. We also found that when training a classifier for cross-genre predictions with word embeddings, it is better to train on a corpus that has instances with a longer length (news) even if they are less similar to the predicted genre (Twitter) than other corpora (YouTube). We believe that the extra length of the training instances contributes to higher quality average embeddings, which makes the classifier more effective.

Of course we must be careful with the last conclusion, since it could also be that the YouTube corpus was of low quality or less similar to the twitter corpus than we thought. One way we could test the suggestion above is to test on news differently. Our current model splits news in tweet-size snippets, classifies them and then takes the most predicted model. However, we might have had better results with the opposite approach; combine a certain amount of twitter messages with the same label, train the model on these larger twitter texts (preferably with news-embeddings), and test on the news corpus. This might be an interesting approach for future research.

References

Gosse Bouma. 2015. N-gram frequencies for Dutch twitter data. *Computational Linguistics in the Netherlands Journal*, 5:25–36.

Felice Dell'Orletta and Malvina Nissim. 2018. Overview of the EVALITA 2018 cross-genre gender prediction (GxG) task. *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA18), Turin, Italy. CEUR. org*.

Rob van der Goot and Gertjan van Noord. 2017. MoNoise: Modeling noise using a modular normalization system. *Computational Linguistics in the Netherlands Journal*, 7:129–144.

Sarah Hamid and Kate Miriam Loewenthal. 1996. Inferring gender from handwriting in Urdu and English. *The Journal of social psychology*, 136(6):778–782.

Tatiana Litvinova, Francisco M Rangel Pardo, Paolo Rosso, Pavel Seredin, and Olga Litvinova. 2017. Overview of the RUSProfiling PAN at fire track on cross-genre gender identification in Russian. In *FIRE (Working Notes)*, pages 1–7.

Erik Tjong Kim Sang and Antal van den Bosch. 2013. Dealing with big data: The case of Twitter. *Computational Linguistics in the Netherlands Journal*, 3(121-134):2013.