Mixing Traditional Methods with Neural Networks for Gender Prediction

Notebook for PAN at CLEF 2018

Rick Kosse, Youri Schuur and Guido Cnossen

University of Groningen, Groningen, The Netherlands
r.kosse,y.m.schuur,g.cnossen.1@student.rug.nl

Abstract In this paper we describe our participation in the PAN 2018 shared task on Author Profiling, identifying author's gender by Tweets and images for English, Spanish and Arabic. We focused only on the textual data and left images out of scope. Our submitted model is a small feed-forward neural network. While in previous work neural networks are often used in combination with word embeddings, our best-performing system used only unigrams as features. In an unofficial run, we show that extracting information from DBpedia can improve the performance. On the PAN 2018 test set our model achieved a score of 0.807, 0.792 and 0.792 for English, Arabic and Spanish respectively. With an average score of 0.797 we conclude that our model is quite robust among all three languages.

1 Introduction

In recent years, author profiling has become increasingly important in daily life. Everyone who publishes text or pictures on social media can be considered an author these days. Where in the past profiling was done by hand, today we take advantage of smart technology. This technology helps us in distinguishing fake news or identifying terrorism threats on social media.

It is interesting how language usage reflects basic social and personality processes and how this reflection can help us to identify gender in social media. Adjacent to this topic, PAN [19] has organized several shared tasks with the the focus on author profiling [15,16,14] in social media. In past series of this shared task, PAN has focused on traits like gender, age, personality type and language variety [14]. This year's author profiling task is to identify the gender of Twitter users.[1] New this year are additional images that (next to the textual Tweets) can help to identify gender. This year's task is for three different languages: Arabic, English and Spanish [13].

We experimented with basic bag-of-words features combined with neural networks, an unusual combination. Although this task provided three different languages we decided to make a single model for all the languages, though trained on the specific language data. We only used the textual data and left images out of scope. In addition, we also experimented with automatically extracting DBpedia features, but the submitted

[1] https://pan.webis.de/clef18/pan18-web/author-profiling.html

system does not include these features due to lack of time. Therefore, this part of the system and the corresponding scores remain unofficial.

In this paper we present a novel approach on the PAN 2018 shared task. We report how our final submitted system works and was optimized. With our submitted system we achieved an average score of 0.797 on the official PAN 2018 test set. For English we achieved a score of 0.807. For both Arabic and Spanish we obtained a score of 0.792.

2 Related Work

In the years that PAN organized the author profiling shared task, many approaches and models haven been submitted. Last year the N-GRAM team won with a straightforward Support Vector Machine (SVM) trained with combinations of character and tf-idf n-grams [2]. A logistic regression with combinations of character, word and POS n-grams finished in second place [8]. In third place, a list of words per variety, learned with an SVM [21]. Noticeable is that they all used simple classifiers as an approach for their classification task in combination with traditional characters and word n-grams.

Some deep learning techniques have been applied in last year's competition, though not with the best results. Word embeddings have been used in combination with a convolutional network [18], scoring 0.78 on the gender task for English. Another CNN deep learning approach achieved a score of 0.74 on the English gender task with traditional tf-idf n-grams combined with word embeddings [17]. The approach of [9] also used word embeddings in combination with character embeddings but with a CNN, RNN, attention mechanism, max-pooling layer, and fully-connected layer. Hereby scoring the best of all the neural network approaches with an average score of 0.813. A different approach was the use of Deep Averaging Networks with character embeddings [6]. To summarize, word/character embeddings were widely used in combination with neural networks, but their results varied a lot.

Another different approach is the cognitive approach by Rangel et al. (2013) based on the neurology studies of Broca and Wernicke about the way users express themselves online [12]. They used Part-of-Speech (POS) Tag frequencies to determine gender differences by examining all kinds of online data (among which Twitter messages [12]). The results showed that men use more prepositions, while women use more pronouns, determinants and interjections. We will also try an approach that uses POS-tags, but we will use them to automatically extract information from DBpedia.

3 Data

The PAN 2018 training set consists of Tweets in three different languages, grouped by Tweet authors, which are labeled by gender. Table 1 shows the number of training instances released by the organization, which are equally distributed over male and female. We divided the data set in training data and test data to develop and optimize our system.

Table 1. Data set PAN 2018

Language	Authors(n)	Train	Test
English	3000	2400	600
Arabic	1500	1200	300
Spanish	3000	2400	600

Since the data was extracted from Twitter, it contained some typical Twitter elements, such as mentions (@username), links, hashtags and excessive use of punctuation. In several previous studies [6,1,7,17,10] all Twitter elements have been removed. We found that replacing them with a dummy value achieved better results then removing them. Furthermore, we tokenized the data by lowercasing. We preprocessed the data step by step. When the scores improved (using our model described in the next section), the method remained, otherwise the method was ignored. The total overview of preprocessing steps, can be found in Table 2.

Table 2. Final number of preprocessing steps applied for all languages.

Preprocessing methods
Lowercasing all Tweets
Replace mentions (@username) with string `username`
Replace hashtags (#...) with string `hashtag`
Replace links (http://...) with string `link`

4 System

4.1 General Model

We decided to submit a feed-forward neural network with traditional sparse n-hot encoding created with the open source library Keras [4]. After a parameter search, the model obtained the best performance with an Adadelta optimizer and a learning rate of 0.22, feeding it with a batch size of 64 and training for 15 epochs. Moreover, the input layer consisted of 100 neurons with a he_uniform weight initialization, using a max norm kernel constraint of 5. Next, a RELU activation function was applied, followed by a dropout layer. During optimization, we found that a relatively big dropout rate of 0.4 outperformed the smaller dropout rates. Finally, the output layer is a single neuron, followed by a sigmoid activation function. Multiple intermediate layers with different neurons were tried as well but did not come close to the score achieved by the smaller model. Therefore, the model was kept to a minimum. The feature set provided to the model was an n-hot encoding of the unigrams.

4.2 Optimization

For optimization we used our Keras model in combination with scikit-learn [11], wrapping the model with the KerasClassifier class[2]. We used the Grid Search functionality, which is a model hyperparameter optimization technique. We provided a dictionary of values and parameters to optimize the accuracy score. Since optimization is rather time consuming, we used a 3-fold cross validation to evaluate each individual model. The outcome described the combination of parameters that achieved the best results. In Table 3 all tested parameters are provided in combination with their best fit.

Table 3. Hyperparameters optimization

Parameter	**Best fit**
Batch size	64
Epoch	15
Dropout regularization	0.4
Weight initialization	he_uniform
Activation function	RELU
Optimizer	Adadelta
Learning rate	0.22
Kernel constraint	maxnorm 5
Number of neurons	100

Due to the popularity of neural networks in combination with word/character embeddings last year [17,9,6] we have conducted experiments with using pre-trained word embeddings in combination with our feed-forward network. However, they did not outperform the score of the model as described above. Therefore, we stayed with our bag-of-word feature approach.

4.3 DBpedia and NNP's

We are also interested in whether we could use DBpedia to improve our feature set. Twitter users often talk about certain topics, but since tweets are short, not much information of these topics is provided. We implement an approach that simply takes these topics and checks if there is a DBpedia page available. If this is the case, we add some of the information of that page to the tweets themselves. Aside from the obvious advantage of providing more data, it also provides a more general representation of certain topics, which is especially beneficial if they do not occur often in the training set.

Unfortunately our submitted system does not include this part, because we were not able to finish it in time. This means that this method was not used to get our official shared task results. Nevertheless, we want to inform you about the process of implementing this into our system and the scores we achieved while running it on our test data.

[2] `https://keras.io/scikit-learn-api/`

For our system, we specifically looked at proper nouns as topics. To extract them, we used the NLTK POS-tagger [5], giving us a number of proper nouns per user. These proper nouns are then used as input for our DBpedia approach, using the DBpedia Lookup service.[3].

This is an online service that can be used to create and look up DBpedia URIs by relating keywords, returning labeled information about the corresponding DBpedia URI. We chose to extract the *description* and *types* DBpedia labels as additional information. The reason behind this choice is that these types and descriptions are the most valuable for the DBpedia extraction as it displays an article's most relevant facts [3]. In the case of the descriptions this is useful, because it contains a lot of additional and general data about a particular topic that is derived from the different articles that form the input of the DBpedia dataset [3]. The types, on the other hand, refer to the conceptual categories in which DBpedia topics can be classified [20]. In this way, proper nouns that refer to a DBpedia page can be generalized over particular categories. An example of the way in which we used proper nouns on tweet level to access DBpedia information is illustrated in Table 4.

The example only contains a short tweet about *Carrie Fisher*, but not much information is given. By including the abstract, the model can learn that she played in Star Wars (which is something males would tweet more often about), while the DBpedia types explicitly return that she was an actor and an artist.

Table 4. Example of the extracted DBpedia information for a given tweet from a female user.

Tweet and Proper Noun
"Rip **Carrie Fisher**, may the force be with you always."
Generated DBpedia URI
`http://live.dbpedia.org/page/Carrie_Fisher`
DBpedia Description
Carrie Frances Fisher (born October 21, 1956) is an American actress and writer. She is best known for her role as Princess Leia in the original Star Wars trilogy $(1977 - 83)$ and Star Wars: The Force Awakens (2015).Fisher is also known for her semi-autobiographical novels, including Postcards from the Edge, and the screenplay for the film of the same name, as well as her autobiographical one-woman play, and its nonfiction book, Wishful Drinking, based on the show. Her other film roles include Shampoo (1975), The Blues Brothers (1980), Hannah and Her Sisters (1986), The 'Burbs (1989), and When Harry Met Sally... (1989).
DBpedia Types
Person, Agent, NaturalPerson, Actor, Artist

[3] `https://wiki.dbpedia.org/lookup`

We have two new feature sources that we can use to retrain our system. We use them in a very straight-forward way, simply adding them to the training data when we create our bag-of-words feature set. In the next section, we will show individual scores of these new features, as well as scores for the combination of these features with the tweets themselves.

5 Results

In this section we describe the results on the training data and the official PAN test data. The results on the training data (both 10-fold CV and test set) are shown in Table 5. On English, we obtain roughly the same results for 10-fold CV and the test set, 0.792 and 0.799. For Spanish and Arabic the results are a bit worse, with similar score for test and 10-fold CV.

Table 5. Accuracies of the feed-forward model on the test set and when using 10-fold CV.

	English	**Arabic**	**Spanish**	**Average**
10 fold CV	0.792 (+/- 0.005)	0.793 (+/- 0.005)	0.783 (+/- 0.006)	0.789
Test set	0.799	0.793	0.773	0.781

The scores of the model on the official PAN 2018 test set are presented in Table 6. We see that the model performs best on English (0.807). Spanish and Arabic score roughly the same with an accuracy of 0.792. Interestingly, for English and Spanish, our model scores higher on the official test set than on our own test set with cross validation, meaning that we did not overfit on the training data. Our average score of 0.797 gave us 5th place in the official shared task results, showing that a feed-forward model in combination with bag-of-words features seems to work quite well for this task.

Table 6. Official results on the PAN 2018 test set.

English	**Arabic**	**Spanish**	**Average**
0.807	0.792	0.792	0.797

Unofficial results from the system that included the DBpedia features are presented in Table 7. We see that the system performs better when the DBpedia types are added to the tweets (0.815 and 0.807). When we add the descriptions to the tweets our system performance drops to a much lower score (0.715 and 0.711). For the scores of our system on only DBpedia descriptions or DBpedia types we can see that that, interestingly, the descriptions score a lot higher than the types. A possible reason for this is that the descriptions contain a lot of data in comparison to the types, which makes classification on this data easier. However, when adding the data to the tweets, the descriptions tend

to overshadow the tweet data, as the descriptions are often longer than the tweets themselves. This makes the system less accurate. On the other hand, the type information, though receiving a lower score individually, is a small but beneficial feature source.

In general, we see an improvement of 1.5 and 1.6% in accuracy for adding the DBpedia types information. This increase should not be underestimated, as 13 out of 23 participants scored between 0.785 and 0.815 for (text-only) English on this shared task. We believe this method can possibly be used to improve other systems as well, for example the winner of last year also used n-gram features. Our current proof-of-concept is only for English, but it can be easily be extended to other languages, provided that the DBpedia lookup service and NNP-taggers are available.

Table 7. Accuracies of the feed-forward model with the DBpedia approach for English on our own test set

Feature Combinations	**Test set**	**10 fold CV**
DBpedia types only	0.580	0.596 (+/- 0.010)
DBpedia descriptions only	0.682	0.674 (+/- 0.006)
Tweets only	0.799	0.792 (+/- 0.004)
Tweets + DBpedia descriptions	0.715	0.711 (+/- 0.009)
Tweets + DBpedia types	0.815	0.807 (+/- 0.003)

6 Conclusion

In this paper we described our approach for the PAN 2018 shared task for identifying author's gender by Tweets. We applied a feed-forward neural network in combination with a simple bag-of-words model, combining new methods with traditional ones. We obtained an average result of 0.797 over three languages and a 5th place in the official shared task rankings. It is remarkable that a small model can achieve such a score, showing that the combination of new methods with traditional ones can work surprisingly well. Interestingly, using pre-trained word embeddings did not work for our model, though we did not perform a large number of experiments. Our model seems to be quite robust, since it obtained similar scores the three different languages. In unofficial experiments, we automatically extracted extra features from DBpedia, getting a 1.5% improvement in accuracy for English. Further optimizing this feature resource could be an interesting topic for future work. Also, to further test its robustness, it would be interesting to apply our model to other languages and different domains.

References

1. Adame-Arcia, Y., Castro-Castro, D., Bueno, R.O., Muñoz, R.: Author profiling, instance-based similarity classification
2. Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., Nissim, M.: N-gram: New groningen author-profiling model. arXiv preprint arXiv:1707.03764 (2017)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: science, services and agents on the world wide web 7(3), 154–165 (2009)
4. Chollet, F., et al.: Keras. `https://keras.io` (2015)
5. Cooper, L., Bird, S.: Nltk: The natural language toolkit (2002)
6. Franco-Salvador, M., Plotnikova, N., Pawar, N., Benajiba, Y.: Subword-based deep averaging networks for author profiling in social media. Cappellato et al.[13] (2017)
7. Kheng, G., Laporte, L., Granitzer, M.: Insa lyon and uni pasauâĂŹs participation at pan@ clefâĂŹ17: Author profiling task. Cappellato et al.[13]
8. Martinc, M., Škrjanec, I., Zupan, K., Pollak, S.: Pan 2017: Author profiling-gender and language variety prediction. Cappellato et al.[13] (2017)
9. Miura, Y., Taniguchi, T., Taniguchi, M., Ohkuma, T.: Author profiling with word+ character neural attention network. Cappellato et al.[13]
10. Oliveira, R.R., de Oliveira Neto, R.F.: Using character n-grams and style features for gender and language variety classification
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
12. Rangel, F., Rosso, P.: Use of language and author profiling:identification of gender and age (2013)
13. Rangel, F., Rosso, P., Montes-y-Gómez, M., Potthast, M., Stein, B.: Overview of the 6th Author Profiling Task at PAN 2018: Multimodal Gender Identification in Twitter. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) Working Notes Papers of the CLEF 2018 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2018)
14. Rangel, F., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. Working Notes Papers of the CLEF (2017)
15. Rangel, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd author profiling task at pan 2015. In: CLEF. p. 2015. sn (2015)
16. Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., Stein, B.: Overview of the 4th author profiling task at pan 2016: cross-genre evaluations. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings/Balog, Krisztian [edit.]; et al. pp. 750–784 (2016)
17. Schaetti, N.: Unine at clef 2017: Tf-idf and deep-learning for author profiling. Cappellato et al.[13] (2017)
18. Sierra, S., Montes-y Gómez, M., Solorio, T., González, F.A.: Convolutional neural networks for author profiling. Working Notes Papers of the CLEF (2017)
19. Stamatatos, E., Rangel, F., Tschuggnall, M., Kestemont, M., Rosso, P., Stein, B., Potthast, M.: Overview of PAN-2018: Author Identification, Author Profiling, and Author Obfuscation. In: Bellot, P., Trabelsi, C., Mothe, J., Murtagh, F., Nie, J., Soulier, L., Sanjuan, E., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 9th International Conference of the CLEF Initiative (CLEF 18). Springer, Berlin Heidelberg New York (Sep 2018)

20. Suchanek, F., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge unifying wordnet and wikipedia. https://hal.archives-ouvertes.fr/hal-01472497/documen (2007)
21. Tellez, E.S., Miranda-Jiménez, S., Graff, M., Moctezuma, D.: Gender and language variety identification with microtc. Cappellato et al.[13] (2017)