# Multilingual Author Profiling using LSTMs
## Notebook for PAN at CLEF 2018

Roy Khristopher Bayot
Teresa Gonçalves

Universidade de Évora
Évora, Portugal
d11668@alunos.uevora.pt, tcg@uevora.pt

**Abstract.** This paper shows one approach of the Universidade de Évora for author profiling for PAN 2018. The approach mainly consists of using word vectors and LSTMs for gender classification. Using the PAN 2018 dataset, we achieved an accuracy of 67.60% for Arabic, 77.16% for English, and 68.73% for Spanish gender classification.

**Keywords:** author profiling, twitter, word vectors, word2vec, LSTM

## 1 Introduction

Communication methods have changed rapidly in the recent years especially with the rise of different social media platforms such as Facebook, Instagram, and Twitter. Aside from exchanges in these social media platforms, there are also platforms such as message boards, question answering sites, and recommendation sites such as Reddit, Quora, Yelp, and Amazon that also make up activity online.

Fake profiles are one of the problems with these online communication models. Incomplete information in someone's profile is also a problem. And thus, analyzing the authorship is one way to take measures with this problem. One component of analyzing the authorship of a text is profiling, may it be determining aspects such as age, gender, or personality.

Our work tries a method on gender author profiling in English, Spanish, and Arabic with twitter text using long short term memory recurrent neural networks and organized as follows: Section 2 covers related literature where it initially discusses previous author profiling endeavors, then followed by methods in PAN, followed by long short term memory recurrent neural networks in conjunction with word vectors. Section 3 describes the author profiling task as well as the dataset. Section 4 describes the methodology and results, beginning with the creation of word2vec vectors, the model, details of the training and then how it was evaluated. Section 5 gives the conclusion and recommendations.

## 2 Related Literature

Earlier works of author profiling were shown by Argamon et al. in [2] for gender, age, native language, and personality classification. The work used various

content-based features such as the 1000 frequent words in the text with high information gain. The work also used style-based features such as the nodes of a taxonomic tree made from systemic functional linguistics [11].

Schler et al. in [33] provides another example of author profiling. It is still centered on gender and age classification and it used stylistic and content features. Parts-of-speech tags, function words, hyperlinks, and non-dictionary words composed the stylistic features while word unigrams with high information gain comprised the content features. These were the features were then used on a Multi-Class Real Winnow for the classification.

### 2.1 PAN Editions

PAN is one of the initiatives at CLEF that has various tasks related to author analysis. It has author identification, obfuscation, and profiling. The author profiling task has been running since 2013, with different aspects to the task during every year.

The focus for PAN 2013 [26] was age and gender profiling. The corpus used then were blogs in Spanish and English. The focus was extended more sources in PAN 2014 [26]. This edition had texts from blogs, reviews, twitter, and social media. The task was again expanded in PAN 2015 [28]. In that year, age and gender classification was also accompanied with regression for personality traits. The personality traits included extroversion, stability, agreeableness, conscientiousness, and openness. However, there was only a twitter corpus on four languages - English, Spanish, Italian, and Dutch. The focus for PAN 2016 [29] was cross-genre evaluation. The idea was to train models on tweets and test them on blogs, reviews, and social media. The languages included during this year were English, Spanish, and Dutch.

The focus for PAN 2017 [27] was on determining the author origin given a specific text aside from gender classification. For instance, an English tweet could come from an author from the US, UK, Canada, Ireland, New Zealand, and Australia. A Portuguese tweet could be from Portugal or Brazil. An Arabic tweet could be from Egypt, Gulf, Levantine, or Maghrebi. And a Spanish tweet could be from Argentina, Chile, Colombia, Mexico, Peru, Spain, Venezuela.

Majority of the approaches is similar to Argamon et al. [2] and Schler et al. [33] wherein features are content-based or stylistic-based. There are also other features that are n-grams based or information retrieval based. The classifiers used are also vary from the use of logistic regression, multinomial Naïve Bayes, liblinear, random forests, Support Vector Machines, and decision tables.

Among some of the variations include Weren et al. [39] where the work also used length features such as number of characters, words, sentences. Their approach also check for infomation retrieval features such as cosine similarity, as well as readability features such as Flesch-Kincaid readability score. Marquardt et al. in [17] also used a combination of content-based features (MRC, LIWC, sentiments) and stylistic features (readability, html tags, spelling and grammatical error, emoticons, total number of posts, number of capitalized letters number

of capitalized words). Maharjan et al. [16] used n-grams with stopwords, punctuations, and emoticons. The work also included the idf count. Villena Román et al. [38] used term vector model representation.

One of the more prominent approaches in the previous editions is that of Lopez-Monroy et al. in [14]. It is prominent in the sense that it worked best for most tasks in most editions. They placed second for both English and Spanish in 2013 where they used second order representation based on relationships between documents and profiles. Another work that placed first for English in 2013 was that of Meina et al. [19] while Santosh et al. in [32] worked well for Spanish. The work of Meina et al. [19] used collocations while Santosh et al. [32] used POS features. The work of Lopez-Monroy et al. in [15] gave the best result with an average accuracy of 28.95% on all corpus-types and languages for PAN 2014. They used the same method same method as the previous year [14].

The work of Alvarez-Carmona et al. [1] used second order profiles similar to previous years. They used it in conjunction with LSA to get the best results on English, Spanish, and Dutch for PAN 2015. The work of Gonzales-Gallardo et al. [10] on the other hand, used character n-grams and POS n-grams that gave the best result for Italian.

In 2016, there had been multiple comparisons since the test genre for the early bird was different from that of the final evaluation, and there had been comparisons with the earlier years as well. However, looking at the final ranking, the top 3 are Busger et al. [23], Modaresi et al. [22], and Bilan et al. in [4]. Individually, the work of Bougiatiotis and Krithara [5] is the top for English while the work of Deneva et al. [9] is the top for Dutch, while Busger et al. [23] and Modaresi et al. [22] are tied for Spanish. Busger et al. [23] used combinations of stylistic features such as function words, parts-of-speech, emoticons, and punctuations signs. The combined this with second order representation and trained their models with SVM. Modaresi et al. [22] used a combination of lexical features with word and character n-grams together with stylometric features as inputs to a logistic regression classifier. Bilan et al. [4] used parts-of-speech, collocations, connective words and various other stylometric features for its classification. Bougiatiotis and Krithara [5] also used stylometric features with character n-grams and the second order representation in conjunction with SVM.

In 2017, although there have been approaches that are more related to deep learning such as RNN [8] and CNN [13], most of the top results were given using SVMs [7]. For instance, the top result came from Basile et al. [3] who used a combination of character and tf-idf n-grams to train an SVM. The second result came from Martinc et al. [18] who used a combination of character, word, and POS n-grams, emojis, sentiments, character flooding, and lists of words per variety as features to a logistic regression classifier. The third best result done by by Tellez et al. [36] also used an SVM.

### 2.2 LSTM and word vectors

Most of the previous approaches hinges on extracting predefined features such as that for style and content. However, a recent trend is to use neural networks to

learn certain filters at run time and use the learned filters to generate a feature representation suitable for classification. This approach need two things - word vectors and the neural network architecture.

Word vectors or word embeddings are needed to be created to represent words in a dictionary. These vectors capture some semantic relation between the words and word2vec is one of the prominent vectors developed by Mikolov in [20] [21]. To create the vectors, random numbers are initially used for words from a dictionary of a corpus such as a Wikpedia dump. Then, by going through the text in the corpus, a word's vector representation is learned by predicting using adjacent words. Getting the vector can be done through either skip grams or continuous bag of words (CBOW). In CBOW, the word vector is predicted given the context of adjacent words while it is the opposite in skip grams. The context words are predicted given a word. The word vectors are then updated after all the predictions are made.

Choosing an architecute comes next after creating word vectors. Among neural network architectures, recurrent neural networks are specifically good for sequences such as text since it uses the previous inputs along with the current input for prediction. This can be shown in the simple recurrent network developed by Jeff Elman in the paper [8]. However, recurrent neural networks usually suffer from vanishing gradient problem especially with long sequences. One way this was dealt with was using long short term memory units which originally proposed by Hochreiter and Schmidhuber in [12]. The idea was to make analog gates that control which information could be stored and which information could be used in remembered. This allowed the propagated errors to be more constant and thus help with the vanishing gradient problem.

An example of using LSTM for classification is that of Rao and Spasojevic in [30]. They used LSTMs for two different datasets - that of customer service and that of political leaning. Their problem was to classify text as either actionable or non-actionable in the domain of customer service, while classifying either Republican or Democrat in terms of political leaning.

Another example is that of Tang et al. in [35] for document classification. In their work, CNNs and LSTMs were used to learn sentence representations and the results were encoded with a gated recurrent network. Their model was used on reviews of IMDB and Yelp.

## 3 Methodology and Results

The figure 1 given below shows an overview description of the system from how the dataset is manipulated before fed into the LSTM and how it is evaluated. The details are described in the following subsections.

### 3.1 Dataset

In the current edition of PAN 2018 [34] for author profiling [25], the task is to predict gender based on text, images, or both. The current dataset has 1500

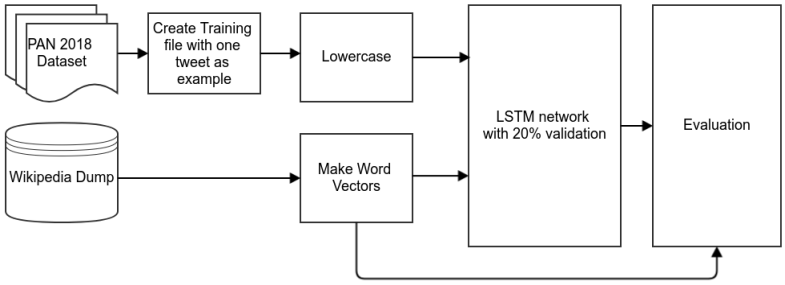


**Fig. 1.** Illustration of the flow of the data from the split, to preprocessing, to feeding to the network, and to the evaluation.

users for Arabic, and 3000 users each for English and Spanish. These are all balanced with an equal number of male and female.

Each user has 100 tweets and 10 images. The images do not necessarily contain an image of a person who is the user but an assorted number of images the user has on the profile. The idea is to train a model to classify a user with tweets alone, or images alone, or both. The model is then submitted to the TIRA server [24] for evaluation over a held-out test set.

### 3.2 Pre-trained Vectors

Word embeddings were created from Wikipedia dumps. The February 05, 2016 wikipedia dump was used for English and Spanish. The English wikipedia dump at that time was 11.8Gb compressed while the Spanish had 2.2Gb compressed. The Arabic wikipedia dump was from March 20, 2018 with about 600Mb compressed. These dumps were then extracted and transformed into lowercase and entries are in one file. The word2vec implementation of gensim [31] was used to generate our own vectors by using the wikipedia text as input. In terms of word2vec parameters, no lemmatization was done, and the window size used was 5. Skip grams instead of continuous bag of words was used as the method to generate the vectors and finally the size of the embeddings chosen was 300.

### 3.3 Preprocessing

Before training the model, we prepared the training file. The training file was done by preprocessing the XML files. Each user has one XML files and the tweets were extracted to form one training example. The examples were all put to lower case. No stop words are removed. Hash tags, numbers, mentions, shares, and retweets were not processed or transformed to anything else. They were retained as is and will correspond to another item in the dictionary of words. The test set from TIRA were also processed in the same manner.

### 3.4 Model

We have a basic model for this experiment that is implemented in Keras [6] with a Theano [37] backend ran on an NVIDIA Tesla K20c GPU. The model mainly consists of an embedding layer, one LSTM layer, and one dense layer.

All words in the training set are turned into number indices that corresponds to a word vector. Each training example will be represented by a sequence of numbers. The sequence length will vary. The total number of indices in the sequence is held at 64 and padding is done to ensure it. We then feed the sequence into the system. Each number will be looked up in the embedding layer and converted to a word vector according to the pre-trained word vectors previously discussed. Then the word vectors passed to an LSTM layer with an output of 32, a dropout value of 0.2, and a recurrent dropout value of 0.5. The output of the LSTM layer is sent to a dense layer with 2 output units and a sigmoid activation function.

Stochastic gradient descent over shuffled mini-batches with the Adam update rule was used for training. Each mini-batch is had 4096 examples. The development set is comprised of 20% of the training set. We also kept the number of epochs to 200 and to provide for early stopping. We saved the best model trained and used it for our test.

### 3.5 Evaluation

When the training finishes and the model is saved, we load the model from the test file and apply it on the tweets per user. After getting a predictions for all the tweets, we used the majority prediction as a final prediction for the user.

### 3.6 Results

We used the model on the full training set and predicted the gender. The confusion matrix of the results are given in tables 1, 2, 3 for Arabic, English, and Spanish respectively.

It gives an accuracy of 81.80% for Arabic, a misclassification rate of 18.20%. It's also 82.44% likely to be an actual male when it predicts males. It's also rougly similar for predicting females, which has 81.18% to be actually female. English accuracy is 85.13%. The misclassification rate is about 14.87%. When it predicts male, it is 86.55% likely to be actually male. When it predicts female, it is 83.82% likely to predict female. Finally, Spanish accuracy is at 75.27% with a misclassification rate of 24.73%. When it predicts male, it's likely to be actually male by 73.51%. When it predicts female, it's likely to be actually female by 77.30%.

However, when this model was applied to the test set in the TIRA servers, the results are lower than the given accuracies. The results of our approach are in table 4. Comparing with the results from other contestants, our approach was 18th globally. We ranked 20 out of 23 for Arabic, 17 out of 23 for English, and 19 out of 23 for Spanish. The highest accuracy achieved for Arabic text was 0.8170,

English text was 0.8221, and Spanish text was 0.8200. The difference between the accuracies are 14.1%, 5.05%, and 13.27% for Arabic, English, and Spanish respectively. English has the closest gap. Perhaps it is also due to the word vectors used. Since the word vectors used came from a bigger resource, 11.8Gb against 2.2Gb and 600Mb of the other languages, it could have contributed to better vectors that were used in the classification problem.

| | Predicted | |
|---|---|---|
| | Male | Female |
| Male | 606 | 144 |
| Female | 129 | 621 |
| | 735 | 765 |

**Table 1.** Arabic confusion matrix over training set

| | Predicted | |
|---|---|---|
| | Male | Female |
| Male | 1248 | 252 |
| Female | 194 | 1306 |
| | 1442 | 1558 |

**Table 2.** English confusion matrix over training set

| | Predicted | |
|---|---|---|
| | Male | Female |
| Male | 1185 | 315 |
| Female | 427 | 1073 |
| | 1612 | 1388 |

**Table 3.** Spanish confusion matrix over training set

## 4 Conclusion and Recommendation

To summarize, we were able to use word vectors together with long short term memory networks as for classification. Our approach is higher than 50% however

| | Accuracy |
|---|---|
| Arabic | 0.6760 |
| English | 0.7716 |
| Spanish | 0.6873 |

**Table 4.** Accuracy results over the test set

it is among one of the lowest in terms of accuracy. We submitted a naive approach to LSTM and there are multiple parameters that still could be explored. Aside from the breadth of hyperparameters, it would also be interesting to see if using a vector for characters instead of words would be useful. This could be an interesting direction since some of the past approaches to classification that worked well has used character ngrams. Another approach could also be a way to incorporate stylometric features to the model.

## References


1. Miguel A Álvarez-Carmona, A Pastor López-Monroy, Manuel Montes-y Gómez, Luis Villaseñor-Pineda, and Hugo Jair-Escalante. Inaoe's participation at pan'15: Author profiling task. In *Working Notes Papers of the CLEF 2015 Evaluation Labs*, 2015.
2. Shlomo Argamon, Moshe Koppel, James W Pennebaker, and Jonathan Schler. Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2):119–123, 2009.
3. Angelo Basile, Gareth Dwyer, Maria Medvedeva, Josine Rawee, Hessel Haagsma, and Malvina Nissim. N-gram: New groningen author-profiling model. *arXiv preprint arXiv:1707.03764*, 2017.
4. Ivan Bilan and Desislava Zhekova. Caps: A cross-genre author profiling system. In *CLEF (Working Notes)*, pages 824–835, 2016.
5. Konstantinos Bougiatiotis and Anastasia Krithara. Author profiling using complementary second order attributes and stylometric features. In *CLEF (Working Notes)*, pages 836–845, 2016.
6. François Chollet. keras. `https://github.com/fchollet/keras`, 2015.
7. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
8. Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
9. Pepa Gencheva, Martin Boyanov, Elena Deneva, Preslav Nakov, G Georgiev, Y Kiprov, and I Koychev. Pancakes team: a composite system of genre-agnostic features for author profiling. *Working Notes Papers of the CLEF*, 2016.
10. Carlos E González-Gallardo, Azucena Montes, Gerardo Sierra, J Antonio Núñez-Juárez, Adolfo Jonathan Salinas-López, and Juan Ek. Tweets classification using corpus dependent tags, character and pos n-grams. In *Proceedings of CLEF*, 2015.
11. Michael Halliday, Christian MIM Matthiessen, and Christian Matthiessen. *An introduction to functional grammar*. Routledge, 2014.
12. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

13. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
14. Adrian Pastor Lopez-Monroy, Manuel Montes-y Gomez, Hugo Jair Escalante, Luis Villasenor-Pineda, and Esaú Villatoro-Tello. Inaoe's participation at pan'13: Author profiling task. In *CLEF 2013 Evaluation Labs and Workshop*, 2013.
15. Adrián Pastor López-Monroy, Manuel Montes-y Gómez, Hugo Jair Escalante, and Luis Villaseñor Pineda. Using intra-profile information for author profiling. In *CLEF (Working Notes)*, pages 1116–1120, 2014.
16. Suraj Maharjan, Prasha Shrestha, and Thamar Solorio. A simple approach to author profiling in mapreduce. In *CLEF (Working Notes)*, pages 1121–1128, 2014.
17. James Marquardt, Golnoosh Farnadi, Gayathri Vasudevan, Marie-Francine Moens, Sergio Davalos, Ankur Teredesai, and Martine De Cock. Age and gender identification in social media. *Proceedings of CLEF 2014 Evaluation Labs*, 2014.
18. Matej Martinc, Iza Škrjanec, Katja Zupan, and Senja Pollak. Pan 2017: Author profiling-gender and language variety prediction. *Cappellato et al.[13]*, 2017.
19. Michał Meina, Karolina Brodzinska, Bartosz Celmer, Maja Czoków, Martyna Patera, Jakub Pezacki, and Mateusz Wilk. Ensemble-based classification for author profiling using various features. *Notebook Papers of CLEF*, 2013.
20. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
21. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
22. Pashutan Modaresi, Matthias Liebeck, and Stefan Conrad. Exploring the effects of cross-genre machine learning for author profiling in pan 2016. In *CLEF (Working Notes)*, pages 970–977, 2016.
23. Mart Busger op Vollenbroek, Talvany Carlotto, Tim Kreutz, Maria Medvedeva, Chris Pool, Johannes Bjerva, Hessel Haagsma, and Malvina Nissim. Gronup: Groningen user profiling. 2016.
24. Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*, pages 268–299, Berlin Heidelberg New York, September 2014. Springer.
25. Francisco Rangel, Paolo Rosso, Manuel Montes-y-Gómez, Martin Potthast, and Benno Stein. Overview of the 6th Author Profiling Task at PAN 2018: Multimodal Gender Identification in Twitter. In Linda Cappellato, Nicola Ferro, Jian-Yun Nie, and Laure Soulier, editors, *Working Notes Papers of the CLEF 2018 Evaluation Labs*, CEUR Workshop Proceedings. CLEF and CEUR-WS.org, September 2018.
26. Francisco Rangel, Paolo Rosso, Moshe Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches. Overview of the author profiling task at pan 2013. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 352–365. CELCT, 2013.
27. Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein. Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. *Working Notes Papers of the CLEF*, 2017.

28. Francisco Rangel, Paolo Rosso, Martin Potthast, Benno Stein, and Walter Daelemans. Overview of the 3nd author profiling task at pan 2015. In L Cappellato, N Ferro, J Gareth, and E San Juan, editors, *CLEF 2015 Labs and Workshops, Notebook Papers*, volume 1391, 2015.
29. Francisco Rangel, Paolo Rosso, Ben Verhoeven, Walter Daelemans, Martin Pottast, and Benno Stein. Overview of the 4th Author Profiling Task at PAN 2016. In Krisztian Balog, Linda Cappellato, Nicola Ferro, and Craig Macdonald, editors, *Working Notes Papers of the CLEF 2015 Evaluation Labs*, volume 1609 of *CEUR Workshop Proceedings*, pages 750–784. CLEF and CEUR-WS.org, September 2016.
30. Adithya Rao and Nemanja Spasojevic. Actionable and political text classification using word embeddings and lstm. *arXiv preprint arXiv:1607.02501*, 2016.
31. Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.
32. K Santosh, Romil Bansal, Mihir Shekhar, and Vasudeva Varma. Author profiling: Predicting age and gender from blogs. *Notebook Papers of CLEF*, 2013.
33. Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, volume 6, pages 199–205, 2006.
34. Efstathios Stamatatos, Francisco Rangel, Michael Tschuggnall, Mike Kestemont, Paolo Rosso, Benno Stein, and Martin Potthast. Overview of PAN-2018: Author Identification, Author Profiling, and Author Obfuscation. In Patrice Bellot, Chiraz Trabelsi, Josiane Mothe, Fionn Murtagh, Jian Yun Nie, Laure Soulier, Eric Sanjuan, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 9th International Conference of the CLEF Initiative (CLEF 18)*, Berlin Heidelberg New York, September 2018. Springer.
35. Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, 2015.
36. Eric S Tellez, Sabino Miranda-Jiménez, Mario Graff, and Daniela Moctezuma. Gender and language variety identification with microtc. 2017.
37. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.
38. Julio Villena Román and José Carlos González Cristóbal. Daedalus at pan 2014: Guessing tweet author's gender and age, 2014.
39. Edson RD Weren, Viviane Pereira Moreira, and José Palazzo M de Oliveira. Exploring information retrieval features for author profiling. In *CLEF (Working Notes)*, pages 1164–1171, 2014.