Using Translated Data to Improve Deep Learning Author Profiling Models

Veenhoven, Robert; Snijders, Stan; van der Hall, Daniël; van Noord, Rik

Published in:
Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)

Using Translated Data to Improve Deep Learning Author Profiling Models

Notebook for PAN at CLEF 2018

Robert Veenhoven, Stan Snijders, Daniël van der Hall, and Rik van Noord

University of Groningen, Groningen, The Netherlands
{r.veenhoven.1,s.m.w.snijders,d.h.a.m.van.der.hall}@student.rug.nl,
r.i.k.van.noord@rug.nl

Abstract In this report on our participation in the PAN shared task on author profiling, we describe our attempt to identify the gender of authors using their posted tweets and images. The data of interest are tweets in the English, Spanish and Arabic languages as well as images. Included in our report is our final submitted system, a bi-LSTM model with attention, as well as an explanation on the less effective solutions we explored. We also detail an approach to obtain more training data, by simply translating the gold standard data of other languages to the language of interest. This proved to be a cheap and robust method for increasing the accuracy of all three languages. Official test accuracy scores are 79.3, 80.4 and 74.9 for English, Spanish and Arabic respectively.

1 Introduction

As social media continues to grow as a source of ever more voluminous and varied data, research into new and more effective ways of analyzing these streams of information continues apace. The text, still images and audio/video files shared over these platforms are the main interest for its users, as it is what attracts them to the platform. Details of the users themselves are often more valuable to the providers: it is what allows them, or their clients, to create advertisements such that they are more likely to influence susceptible users. Since this user information is not always provided by the people involved, methods to uncover age, gender, education and other variables using the posted content provided by them as data are needed. Research into this field pre-dates modern social media; more conventional private weblogs prompted [6] to predict their authors' ages based on the posted text in the very year Twitter was founded, now more than a decade ago.

The Author Profiling shared task [25] organized as part of the PAN 2018 evaluation lab [32] provides an opportunity to tackle gender identification for Twitter users. Previous research and iterations of this shared task [26] have shown that tweets of various languages can be used to identify the gender of its author. The winner of the previous incarnation of the PAN Author profiling shared task, [2], reported accuracy scores between 80.1 and 84.5 for four languages.

We present our efforts to create a model that performs this identification for English, as well as Spanish and Arabic tweets. We implement both a CNN and a bi-LSTM model

with pre-trained word embeddings for all three languages, showing that a bi-LSTM is a better choice for this task. Moreover, we show that it is possible to (cheaply) obtain extra training data, by simply translating the gold standard data of other languages to the language of interest. Our accuracy scores on the official test set are 79.3%, 80.4%, and 74.9% for English, Spanish and Arabic.

2 Model Architecture

We propose two neural network models for the author profiling task, where one model is based on a Convolutional Neural Network (CNN) and one model is based on a bidirectional Long Short-Term Memory network (bi-LSTM), which belongs to the family of Recurrent Neural Networks (RNN). We decided to focus on these models since it seems that these deep learning models are not explored to their full potential in the author profiling task of 2016 and 2017, as most teams focused on Support Vector Machines (SVM) [26,27]. Some exceptions to this trend were [15], who used a bi-directional GRU with an attention mechanism to attain 11th place in the 2017 gender identification task, and [20] who came in 3rd in the same task with a combined RNN/CNN approach, though they also did not use an LSTM-based model.

Though these approaches are sparse in this sub-area, they have had many successes in other areas. For example, deep learning models were prevalent at the SemEval 2017 shared task on sentiment analysis, the WASSA 2017 shared task on Emotion Intensity and the CoNLL 2017 Shared Task, where the top teams outperformed SVMs using deep learning models in the form of CNN and RNN [26,21,39]. This is why this paper will focus on implementing these two methods for this task.

2.1 Preprocessing

In order to provide our models with the correct input, we first have to preprocess the data. First a tweet gets tokenized using the NLTK TweetTokenizer [4]. Then we remove all punctuation characters. In the third step, we simultaneously lowercase all words and remove all stopwords using the NLTK stopword corpora. In the fourth and final step, all hyperlinks and usernames are replaced with the tokens `URL` and `USER` respectively.

2.2 Embeddings

Both our approaches are dependent on pre-trained word embeddings. For each user in the training data, we convert the tweets to a vector representation, where each word is represented by its index. These are then padded, so that the vector of every user has the same length. To initialize our embedding layer we use all the unique words in our training data, which leads to a matrix of size $s \; x \; d$, where s is the amount of unique words in the training data and d is the embedding space dimension. To initialize the weights, we use the GloVe Twitter embeddings for English, which were created using 2 billion tweets and 1.2 million unique words [22]. For Spanish we use the word embeddings provided by [16] and for Arabic we train the embeddings ourselves on gathered tweets using Word2Vec with default parameters [18]. We train these using 223 million tweets

and 3 million unique words. In all three languages a word is represented by a vector of dimension 200 and therefore we set our d as 200.

2.3 CNN

[14] showed that a simple CNN with little hyperparameter tuning can achieve good results on sentence classification, even though CNN models were initially mainly successful in computer vision. Since then, CNNs have been successfully applied in many NLP areas, including, among others, sarcasm detection [19], text categorization [12], dependency parsing [38] and semantic parsing [36]. CNNs have also been tried before for gender classification, though they were not very successful [30,28].

Our model uses frozen embedding weights and does not update these during training, since [14] reported that the performance is poor when these are trained at the same time as the CNN training. This is also confirmed by [13] who argued that this would lead to overfitting. Although training the embedding layer from the start did not result in better scores for our model, we didn't experiment with updating the weights after a specified amount of epochs. This approach of gradual unfreezing can result in better performance as showed by [11].

Next, we employ the embedding layer on the padded tweets and use this as input for the convolutional layer. We use a one dimensional convolutional layer, where each convolution uses a filter matrix of size f x d where f is the amount of words in the convolution and d is the embedding space dimension. On each window of the user's tweets, we create a feature map by using this filter matrix. To do this, we use Rectified Linear Unit (ReLU) activation. We then use a max-pooling operation on the feature map of each filter to extract the maximum value. This value can be described as the most important feature of the feature map. The outputs of each feature map are then concatenated into a new feature vector that is passed to a dropout layer. Dropout layers are used to prevent overfitting by randomly setting weights to zero [31]. This is then fed to a softmax layer, which returns the probability that the user is male and the probability that the user is female.

2.4 RNN

As our RNN model, we propose a bidirectional LSTM model that uses an attention mechanism as proposed by [35]. An advantage of RNN models is that it handles input sequentially, taking word order into account, while this information is lost in a CNN model. LSTM models use cell states to handle new information [10]. This way, an LSTM can use its input and forget gate to learn and forget selectively for shorter and longer periods of time, without modifying all the existing information.

For each word in a sentence, the LSTM model combines its previous hidden state and the current word's embedding weight to compute a new hidden state. This information is then fed to the output gate, which selects the most informative information to output. We also use a bidirectional model, which gives additional information over normal RNN and CNN models [29].

As input for our bi-LSTM, we use a padded vector representation of the tweets of a user based on the word index. For our model, we first use the same embedding

layer as proposed in our convolutional model. We apply a zero-padding algorithm so that every user has the same matrix size, which is in line with the method proposed by [14]. To make sure that our embedding layer would not read these zeros as a unique word integer, we use the *mask_zero* parameter of the embedding layer. The embedding weights are frozen, since this increased the performance of the system. After applying dropout, we use an attention mechanism as proposed by [35] to emphasize the most informative words in the combined tweets of a user. This is then fed to a softmax layer, which returns probabilities for male or female.

An overview of our system can be found in Figure 1. The same figure can also be used for our CNN, where the bi-LSTM model is replaced by a convolutional model.

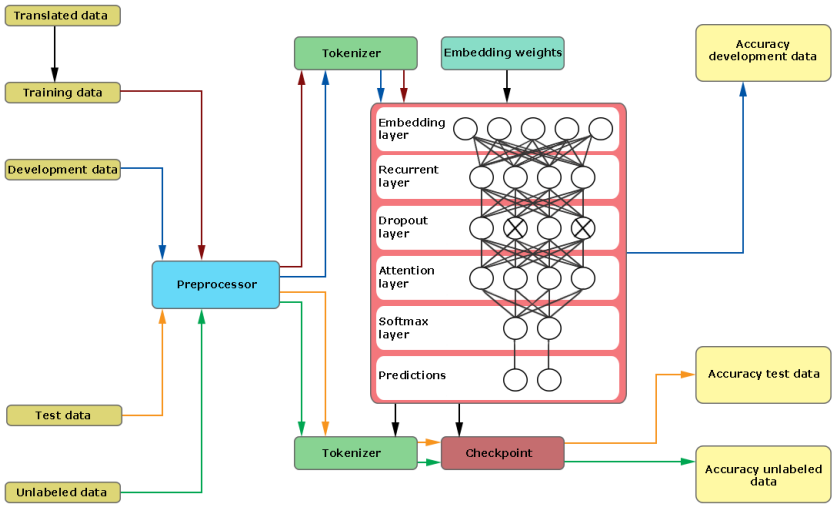

Figure 1. Model architecture and usage of our bi-LSTM model.

3 Optimizing and Training

3.1 Grid search

To optimize the parameters of our CNN and RNN, we run a grid search for multiple parameters. The list of values that we use for the grid search of our CNN can be found in Table 1. For our bi-LSTM, we use the values in Table 2. To do this, we randomly select 20% of our training set and use this for development. For combinations that achieve very similar results, we run our system multiple times and take the average score. This leads to our optimized CNN and RNN.

Table 1. Grid search function names and values for our convolutional model. Best parameter settings in bold.

Function name	Grid values used
Amount of filters	100, **128**, 200
Remove words that only occur once	**True**, False
Training epochs	5, **10**, 20
Batch size	32, **50**, 64
Dropout	0.2, 0.3, **0.5**
Learning rate of Adam optimizer	**0.0001**, 0.001, 0.1

Table 2. Grid search function names and values for our bi-LSTM model. Best parameter settings in bold.

Function name	Grid values used
Nodes	**100**, 200
Remove words that only occur once	**True**, False
Training epochs	10, **20**, 30
Batch size	20, **32**, 50, 64
Dropout	0.2, 0.3, **0.5**
Learning rate of Adam optimizer	**0.0001**, 0.001, 0.1

Our optimized RNN uses 100 units, a dropout rate of 0.5, a batch size of 32, the Adam optimizer in combination with a learning rate of 0.0001 and we remove words that only occur once. The latter is is in line with 2017's top performing system, who found that this removal has a strong positive effect [2]. This does not only lead to a better score of our system, but it also reduces the vocabulary size by more than 50%. Lastly, we train our model for 20 epochs.

3.2 Adding training data

[1] showed that for a prototypical natural language task, the performance can benefit significantly from larger training sets. This is also often observed in shared tasks, where teams try to gather additional training data by, for example, adding data sets of previous years. This method often works well, for example last years winner [2] used this approach in the PAN 2017 author profiling task and [3] used this in the SemEval 2017 sentiment analysis task. Since we already use the PAN 2017 set as test data, this was not an option for our system. The older PAN sets do not contain Arabic, so therefore we decided against using these sets.

Another approach that is often used is applying distant supervision on unlabeled gathered datasets, using the method of [8]. This way, large amount of data sets can be classified using noisy labels for each class. This approach is followed by [7], who scored first on all five subtasks in SemEval 2017's sentiment analysis task.

We employ a lesser known method to obtain more training data. Our approach tries to collect more training data by simply translating the gold standard training data of the other languages. This approach differs from many other data augmentation methods, since the label is now gold standard while the data itself is noisy. A benefit that is only applicable on this shared task is that the provided images do not have to be translated.

The disadvantage of translated data, obviously, is that automatic translation is noisy, especially on informal Twitter data. For example, incorrectly spelled words, deliberately or not, are often not translated at all. Moreover, some language specific phenomena are not translated correctly, which can lead to important information being left out of the translation. On the other hand, other distant supervision methods are also noisy. This approach at least has the advantage that there was manual annotation involved at some point (though for a different language) and that current machine translation software (involving English) is of quite high quality.

The conversions that we make are showed in Table 3. We decide not to use all conversions, since the remaining conversions are of low quality. For example, using a machine to translate Spanish tweets to Arabic leads to more words being Spanish than Arabic in the translation. Adding more data obviously only helps if the data quality is not poor [34]. To translate the data sets we use Google Translate [9].

Since this library cannot handle emojis, we give each emoji a unique identifier that is not translated. After we translate the text of a user, we then replace the unique identifier with the corresponding emoji. This translated data is then merged with the original training data after the training development split, so that the translated silver data is never used as validation.

Table 3. Conversions that we make to add more training data.

From	To
English	Spanish
English	Arabic
Spanish	English

It is interesting to look at the most informative words for male and female Twitter users, to see whether the translations would capture these words. This is shown in Table 4. We see that the most informative words are not individual topics (for example sports and shopping), but more related to how man and women refer to things and each other in general (e.g. mate, fine, bro, cute, lady). These references are usually very specific and might not translate well to other language that have their own conventions.

Table 4. Most informative words for both male and female applying TF-IDF.

Male	mate	gay	game	fine	fuck	bro	years	apple	wife	man
Female	women	thank	woman	love	cute	lady	so	xx	ppl	xxx

3.3 Training

Both our optimized models are then used for the final training runs. As our test set, we use parts of the PAN 2017 author profiling data set [26]. Since most of the 2017 corpus is also used as the 2018 training set, we extract the users that are not used for the 2018 set and use these for testing. This leads to 898 tweets for Arabic, 599 tweets for English and 1200 tweets for Spanish. Our development set for Arabic, English and Spanish contain 20% of the original training data and these are randomly selected. These contain 300, 600 and 600 tweets respectively.

For each language, we run a model twice, once with added translated training data and once with only the original training data. We train each combination three times, of which we average the development and test scores. For our submission, we use the saved model with the lowest averaged validation loss based on the development and test set.

3.4 Images

In addition to the tweet text, the PAN 2018 dataset also includes 10 images for each user. As could be expected, these images vary wildly in content, ranging from depictions of text to landscapes. While new to the PAN Author profiling shared task, images have been used for author gender identification before, for example, [37] used a combination of image type and image content to classify author gender with an average accuracy of 0.719. One type of picture classification for which a considerable number of tools are already available is face recognition, making implementation relatively straightforward. Faces are also a prevalent feature in tweeted images, with [33] finding people depicted in half of a random set of tweets, prompting us to choose this particular aspect of the images to use as a feature.

The PAN 2018 test set contained 10 images for each author, for a total of 30,000 images per language. The images are first processed using the face recognition scripts of [24], itself a reimplementation of the age and gender detection network created by [17]. The images that had at least one face in it were selected using a Haar feature-based classifier found in OpenCV [5]. This resulted in a subset totaling around 8,300 images per language. Each image is then processed to obtain the number of faces present in each, as well as the estimated gender of each face as found by the [24] script. Processing the images results in counts for the number of male and female faces found in all the images of each author. The added information thus consists of two vectors, each containing one integer for each author: the total number of male faces and the the total number of female faces found in that author's images.

The number of faces found in the images did vary along gender lines: while the 10 images of the average female contained 1.47 female and 1.57 male faces, those of their male counterparts contained an average of 0.82 female faces and as many as 2.91 male faces. This tells us that this feature, though straightforward, can supply the algorithm with valuable information.

The face vectors are used with a standard SVM to gauge their effect on gender identification accuracy, comparing and combining it with the standard text dataset as well as the translated variant. For images only, we obtained accuracy scores between 65 and

69. Though this is considerably better than chance-level, and perhaps a good score considering the amount of data available per user and the simplicity of our model, it is not close to the performance of the text-only models. When combining the image features with the textual features we, unfortunately, did not see an improvement in accuracy. The image features were therefore left out of our final submitted models.

4 Results

As mentioned previously, we created our own development and test set to evaluate our models on. The comparison of our CNN and RNN can be seen in Table 5. We see that our bi-LSTM RNN model clearly outperforms the CNN, obtaining a higher score for each language and test set. Therefore, we use our RNN as official model in the shared task, and will report only the scores of the RNN in the rest of this section. Our scores for English are competitive with the state-of-the-art (83.3 in [2]), however, for Spanish and especially the Arabic test set our model does not reach such a high score.

Table 5. Comparison of our RNN and CNN baseline models.

	English		**Spanish**		**Arabic**	
	Dev	**Test**	**Dev**	**Test**	**Dev**	**Test**
RNN	82.7	80.9	78.5	76.0	77.9	68.3
CNN	78.7	75.5	71.8	65.5	74.3	65.3

Another objective of this paper was to determine whether using translated gold standard data can improve performance. Table 6 shows the performance of the RNN model with and without using translated data for each language. We see that for each language, simply adding the translated data results in a higher score for both our dev and test set. The gains are quite considerable, ranging between 0.5 and 1.5%. This shows that a simple method can already increase performance. Also, it should be applicable on lots of other tasks and domains. However, we are aware of the main disadvantage of this approach; it can only be applied if there is manually annotated gold standard data available for multiple languages.

Finally, we present the accuracy scores of our submitted model on the unseen test data provided by the PAN organisation, which can be found in Table 7 below.

Table 7. Accuracy scores of our Bi-LSTM model on the PAN test data set, as retrieved from the Tira system [23].

Language	**Accuracy**
English	79.3
Spanish	80.4
Arabic	74.9

Table 6. Impact of using translated gold standard data. Reported are accuracies of the RNN model on our dev and test sets.

Model	Dev	Test
English	82.7	80.9
English + translated	83.3	82.1
Spanish	78.5	76.0
Spanish + translated	79.3	77.3
Arabic	77.9	68.3
Arabic + translated	78.3	68.8

When comparing the results obtained on the PAN test set and on our own test set, it shows that our model performed slightly worse on the English language with an accuracy score of 79.3% on the PAN data and 82.1% on our own data (-2.8%). Though for the other two languages, an increase in accuracy can be noted between the PAN test set and our own. Our best scoring language on the unseen test data is Spanish, with an accuracy score of 80.4, while it was 77.3 on our own test set (+3.1). For Arabic, we obtained the biggest increase in accuracy scores, with 74.9 on the PAN and 68.8 on our own data (+6.1).

5 Conclusion

In this paper, we presented our system for the 2018 PAN author profiling shared task. The main idea we put forward is the creation of additional training data by translating the data from one language to the other. We explored the different directions in which the translations could be made, but found that resulting translations that do not involve English were too noisy to be considered in the training phase of our models. Nonetheless, it is clear that the addition of the right translated data to our models training data can lead to higher accuracy scores for all three languages. We also compared bi-LSTM to CNN models, finding the bi-LSTM models are better suited for this task, with official final scores of 79.3, 80.4 and 74.9%, for English, Spanish and Arabic.

References

1. Banko, M., Brill, E.: Scaling to very very large corpora for natural language disambiguation. In: Proceedings of the 39th annual meeting on association for computational linguistics. pp. 26–33. Association for Computational Linguistics (2001)
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. Baziotis, C., Pelekis, N., Doulkeridis, C.: Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis pp. 747–754 (01 2017)
4. Bird, Steven, E.L., Klein, E.: Natural Language Processing with Python. O'Reilly Media Inc. (2009)
5. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)

6. Burger, J.D., Henderson, J.C.: An exploration of observable features related to blogger age. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs. pp. 15–20. Menlo Park, CA (2006)
7. Cliche, M.: Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. CoRR abs/1704.06125 (2017), http://arxiv.org/abs/1704.06125
8. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision (2009)
9. Han, S.: Googletrans 2.2.0 : Free google translate api for Python. (2017), https://pypi.org/project/googletrans/, [Online; accessed April 2018]
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
11. Howard, J., Ruder, S.: Fine-tuned language models for text classification. CoRR abs/1801.06146 (2018), http://arxiv.org/abs/1801.06146
12. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 562–570. Association for Computational Linguistics, Vancouver, Canada (July 2017), http://aclweb.org/anthology/P17-1052
13. Karpathy, A., Joulin, A., Fei-Fei, L.F.: Deep fragment embeddings for bidirectional image sentence mapping. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 1889–1897. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5281-deep-fragment-embeddings-for-bidirectional-image-sentence-mapping.pdf
14. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
15. Kodiyan, D., Hardegger, F., Neuhaus, S., Cieliebak, M.: Author profiling with bidirectional rnns using attention with grus: notebook for pan at clef 2017. In: CLEF 2017 Evaluation Labs and Workshop–Working Notes Papers, Dublin, Ireland, 11-14 September 2017 (2017)
16. Kuijper, M., Lenthe, M., Noord, R.: Ug18 at semeval-2018 task 1: Generating additional training data for predicting emotion intensity in spanish. In: Proceedings of The 12th International Workshop on Semantic Evaluation. pp. 279–285 (2018)
17. Levi, G., Hassncer, T.: Age and gender classification using convolutional neural networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 34–42 (June 2015)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
19. Mishra, A., Dey, K., Bhattacharyya, P.: Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 377–387. Association for Computational Linguistics, Vancouver, Canada (July 2017), http://aclweb.org/anthology/P17-1035
20. Miura, Y., Taniguchi, T., Taniguchi, M., Ohkuma, T.: Author profiling with word + character neural attention network. Cappellato et al.[13] (2017)
21. Mohammad, S.M., Bravo-Marquez, F.: Wassa-2017 shared task on emotion intensity. arXiv preprint arXiv:1708.03700 (2017)
22. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
23. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury,

A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014)
24. Pressel, D.: Rude carnie: Age and gender deep learning with tensorflow (2017), available at https://github.com/dpressel/rude-carnie
25. 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)
26. 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)
27. 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)
28. Schaetti, N.: Unine at clef 2017: Tf-idf and deep-learning for author profiling. Cappellato et al.[13] (2017)
29. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11), 2673–2681 (1997)
30. Sierra, S., Montes-y Gómez, M., Solorio, T., González, F.A.: Convolutional neural networks for author profiling. Working Notes Papers of the CLEF (2017)
31. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)
32. 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)
33. Thelwall, M., Goriunova, O., Farida, V., Simon, F., Anne, B., Jim, A., Amalia, M., Emma, S., Francesco, D.: Chatting through pictures? a classification of images tweeted in one week in the uk and usa. Journal of the Association for Information Science and Technology 67(11), 2575–2586 (2016), https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23620
34. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2016)
35. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1480–1489 (2016)
36. Yih, W.t., He, X., Meek, C.: Semantic parsing for single-relation question answering. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 643–648 (2014)
37. You, Q., Bhatia, S., Sun, T., Luo, J.: The eyes of the beholder: Gender prediction using images posted in online social networks. In: 2014 IEEE International Conference on Data Mining Workshop. pp. 1026–1030 (Dec 2014)
38. Yu, X., Vu, N.T.: Character composition model with convolutional neural networks for dependency parsing on morphologically rich languages. In: Proceedings of the 55th Annual

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 672–678. Association for Computational Linguistics, Vancouver, Canada (July 2017), http://aclweb.org/anthology/P17-2106
39. Zeman, D., Popel, M., Straka, M., Hajic, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., et al.: Conll 2017 shared task: multilingual parsing from raw text to universal dependencies. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies pp. 1–19 (2017)