








# Gender Identification in Social Media Using Transfer Learning

Aquilino Francisco Sotelo<sup>1</sup> , Helena Gómez-Adorno<sup>2</sup>  ,  
Oscar Esquivel-Flores<sup>2</sup> , and Gemma Bel-Enguix<sup>3</sup> 

<sup>1</sup> Posgrado en Ciencia e Ingeniería de la Computación, UNAM, Mexico City, Mexico  
[aquilino@comunidad.unam.mx](mailto:aquilino@comunidad.unam.mx)

<sup>2</sup> Instituto de Investigación en Matemáticas Aplicadas y en Sistemas, UNAM,  
Mexico City, Mexico

[{helena.gomez,oscar.esquivels}@iimas.unam.mx](mailto:{helena.gomez,oscar.esquivels}@iimas.unam.mx)

<sup>3</sup> Instituto de Ingeniería, UNAM, Mexico City, Mexico  
[gbele@iingen.unam.mx](mailto:gbele@iingen.unam.mx)

**Abstract.** Social networks have modified the way we communicate. It is now possible to talk to a large number of people we have never met. Knowing the traits of a person from what he/she writes has become a new area of computational linguistics called Author Profiling. In this paper, we introduce a method for applying transfer learning to address the gender identification problem, which is a subtask of Author Profiling. Systems that use transfer learning are trained in a large number of tasks and then tested in their ability to learn new tasks. An example is to classify a new image into different possible classes, giving an example of each class. This differs from the traditional approach of standard machine learning techniques, which are trained in a single task and are evaluated in new examples of that task. The aim is to train a gender identification model on Twitter users using only their text samples in Spanish. The difference with other related works consists in the evaluation of different preprocessing techniques so that the transfer learning-based fine-tuning is more efficient.

**Keywords:** Author profiling · Natural Language Processing · Transfer learning · Classification

## 1 Introduction

Author profiling (PA) is a Natural Language Processing (NLP) task that aims to determine the characteristics of the author(s) of a given text, such as their gender, age, emotional state, personality, among others. AP can be performed on formal and informal textual sources. Formal texts have a certain structure and

---

This paper has been partially supported by the PAPIIT-UNAM projects 401219, TA100520, IA-104720, and CONACyT project A1-S-27780.

© Springer Nature Switzerland AG 2020  
K. M. Figueroa Mora et al. (Eds.): MCPR 2020, LNCS 12088, pp. 293–303, 2020.  
[https://doi.org/10.1007/978-3-030-49076-8\\_28](https://doi.org/10.1007/978-3-030-49076-8_28)

follow rules while informal texts do not follow rules and are not standardized. A good example of the latter are social networks.

The writing style on social media has special features [10] that make NLP tasks extremely complex processes: the abbreviation rules are not always followed, different use of punctuation marks, new characters are included such as # (*hashtag*), use of the sign @ to mention users, etc.

Given the importance and the enormous amount of information that is produced daily in social media, it is necessary to have computational methods that allow us to automatically analyze the information generated in these networks.

With the information that people publish and consume in their social media, companies can profile their clients and governments can improve security procedures, for example, identifying potential cases of pedophilia, virtual kidnappings, among others. In fact, the providers of these services already profile users, for example Twitter aims to know the patterns of use and personalization of content. For these reasons, the aim of this work is to develop an automatic gender identification model of Twitter users using transfer learning techniques. We also measure and evaluate the impact of text preprocessing on the accuracy of the author profiling model.

The work is presented in 5 sections, including this introduction. Section 2 describes the methods to carry out feature extraction and the machine learning algorithms typically used in AP, in Sect. 3 we introduce the concepts of transfer learning and explain the architecture used in this work, in Sect. 4, the methodology for author profiling and experimental results is presented. The conclusions of this paper are enunciated in the Sect. 5.

## 2 Related Work

Several supervised learning techniques were used to model author profiles in different text sources. Supervised learning classifiers employ a set of input-output pairs, through which a decision function is learned that associates a class label with a new data within the established classes. Author profiling (AP) consists in identifying the demographic features of the author of a text [6]. These features are those that describe the author in terms of gender, age, level of study, nationality, socio-economic level, among others. So it can be concluded that AP is a multiclass classification problem.

The use of supervised learning algorithms for AP is shown in [15]. *Decision Functions* are a technique to perform a binary classification, whose training consists of finding decision functions from input-output pairs. *Logistic Regression* is used for multiclass classification problems to predict the probability that the data belong to one or another class. *Support Vector Machines* is a technique used in the context of the AP for binary classification; data are linearly separable by several planes. *Neural Networks* are another resource for AP; the goal of the method is to approximate a function  $g(\cdot)$ , represented by the neural network, to a function  $f(\cdot)$  as much as possible. This approximate function is the one used to classify. *Convolutional Networks* represent an important tool for AP, since they are trained with large sets of information, in addition to setting a feature extractor.

## 2.1 Features Extraction

Analyzing in detail the large amount of information currently generated in the form of written texts is very complicated. Therefore, it is of interest to create representations of these documents, that is, to obtain their representative characteristics. The features obtained from a text are specific terms that allow analyzing and extracting useful patterns or knowledge from analyzed documents. In the past, this task was performed by linguists, limited to a little thorough manual processing. However, with the advance of science and technology, the methods for the extraction of characteristics changed. Some text representation schemes are:

1. **Bag of words.** In order to deal with complete documents it is necessary to use a computationally viable structure. To fulfil this, we see the documents as strings [7]. Let  $S = s_1, s_2, \dots, s_k$  be a string, where a word is a substring of  $S$  of length 1, which can refer to: an item in the text, an item in lowercase or uppercase, the word with its part of speech label (POS), word lemma, any other variant of the word.
2. **N-grams:** Let  $S = s_1, s_2, \dots, s_k$  be a string. The N-grams are defined as substrings of  $S$  of length  $N$ . The 1-gramas are called unigrams, the 2-gramas are called bigrams, and so on. There are two types of N-grams, those of words and those of characters. Word N-grams refer to continuous N-words in the document. Instead, character N-grams refer to the N-characters within the word limit without spaces.
3. **Syntactic N-grams:** Syntactic N-grams try to capture the linguistic structure of a text by organizing the words into nested components in order to show through arrows which words depend on others.

## 2.2 Weighting Schemes

To obtain a representation of a document, a preprocessing is carried out to see it as a vector. Each dimension of the vector stands for a feature of the document. Each feature is represented by assigning some weight according to its relevance, this process is called *weighting scheme*. The most relevant are described below:

1. **Boolean model and Term Frequency (TF):** There are some very intuitive ways to assign weight, such as identifying whether a term appears or not, counting how many times a term appears in a text and assigning a weight to each term depending on the number of occurrences it has.
2. **Inverse Document Frequency:** To treat high frequencies of certain words (due to the context they are constantly repeated), the weight of the Term Frequency (TF) is reduced by means of the Inverse Document Frequency (IDF). This compensates the weight depending on the appearance of the word in many documents or not. The Inverse Document Frequency of a term  $t$  is defined with total frequency in the collection with the expression:

$$idf_t = \log(N/df_f).$$

3. **TF-IDF:** It is the product of the Term Frequency (TF) by the Inverse Document Frequency (IDF). Its purpose is to provide a measure that expresses the relevance of words in such a way that it is possible to distinguish between those that describe the document and those that do not. To assign a weight to the words in a document, the frequency of the words is calculated and, in the total of documents, the weight is calculated with the following expression

$$tf - idf_{t,d} = tf_{t,d} \times idf_t$$

4. **Word embeddings:** A different method for weighting schemes are word vectors (*word embeddings*), that use two main approaches: discrete and distributional. The idea of *discrete approach* is to represent a word in a vector of dimension  $n$  with 1's and the others with 0's; these are also known as *one-hot vectors*, where  $n$  is the number of words in the vocabulary. The *distributional approach* takes into account the similarity between the vectors themselves, when a word appears in a text its context is the set of words that appear near it (a fixed size window). This builds a dense vector for each word, making it similar to the word vectors. The most used methods with this technique are *Word2vec* [13] and *GloVe* [14].

### 3 Transfer Learning

Transfer learning is a subfield within machine learning that has been studied for more than three decades [2]. It tackles the ability to take advantage of pre-existing data sets when you want to learn from new data. One method that has proven to be effective for obtaining knowledge is the pre-training technique with large amounts of previously available data and the subsequent fine tuning of the pre-trained model based on data from new tasks [5]. This pre-training is also known as *few-shot learning*. In transfer learning, first it is trained a neural network on a given data set and a specific task, then the features learned by the network are reused, transferred to a second network to be trained in another task and a different data set.

The transfer learning technique consists in taking advantage of the weights of an already trained neural network and adjust them to solve other tasks with only few examples [16,17]. The types of strategy to perform transfer learning with a new data set are:

- **Fixed feature extractor:** A pre-trained neural network is taken and the last fully connected layers are removed, then the features are extracted with a fixed extractor for a new dataset. Finally, a linear classifier (for example SVM) is trained for the new dataset.
- **Fine tuning.** In addition to replacing and re-training the classifier, the weights of the pre-trained network are adjusted by continuing back propagation.
- **Pre-trained models.** It consists of taking advantage of the final control points of the neural network already trained to make adjustments.

To know the type of transfer learning that is more suitable to be carried out, the following criteria are taken into account:

- The new dataset is small and similar to the original dataset so it can lead to overfitting the model, fine tuning does not work here. Therefore it is best to train a linear classifier.
- The new dataset is large and similar to the original dataset. As there is more information, the risk of overfitting is low, therefore fine tuning can be applied.
- The dataset is small but very different from the original dataset. Because there is little data, it is best to train a linear classifier. As it is different from the original dataset, it may be very different from its specific characteristics.
- The new dataset is large and very different from the original. As there is enough data and they are different from the original it is best to apply the strategy of pre-trained models.

According to [4] there are two strategies for transfer learning for text:

- **Feature based:** it consists on pre-training vectors that capture the additional context through other tasks. New vectors are obtained for each layer that are then used as characteristics, concatenated with the word vectors or with the intermediate layers, an example of this is *ELMo* [12].
- **Fine tuning:** It consists on pre-training some architecture in an objective language model before refining it for a supervised subsequent task, introducing a minimum number of specific parameters of the task, and training in subsequent tasks simply by refining the pre-trained parameters [8].

In our case, we have a relatively small corpus to perform author profiling, so our strategy is to use the *Fixed feature extractor* technique. Below we describe the algorithm for extracting features.

### 3.1 Universal Sentence Encoder

Here we describe the transfer learning based algorithm we used to extract features for performing author profiling, which is called *Universal Sentence Encoder* (USE) [3]. Although this method is not designed specifically to perform author profiling, it has certain characteristics that can be used for this task. The *Universal Sentence Encoder* encodes text in high-dimensional vectors so that it can be used for text classification, semantic similarity, clustering, and other natural language tasks. The model is trained in a variety of text data sources and a variety of tasks in order to dynamically accommodate a wide variety of natural language comprehension tasks. Specifically, USE has two models to encode documents in word vectors, one makes use of the architecture based on averages called *Deep Averaging Network* (DAN) [9], while the other is based on a convolutional neural network for document classification [11]. These architectures are detailed below for a better understanding:

1. **Deep Averaging Network:** This architecture works in three steps:
  - Average the vectors associated to a token sequence
  - Pass the average through one or more layers of a Feed Forward
  - Make the linear classification in the last layer
2. **Convolutional Neural Network:** This type of network, receives a a document as a sequence of vectors in the input layer. It applies the average sampling (*average pooling*) to convert the word vectors into a document vector representation of fixed length. Document vectors are obtained after averaging the word vectors through one or more feed forward layers with fully connected layers.

For this work we used a USE model trained in multiple tasks across 16 languages, including Spanish. USE receives as input a text of variable length in any of the languages in which it was trained and the output is a vector of 512 dimensions. The USE model we use is available from the *TensorFlowHub*<sup>1</sup> page and can be freely downloaded. In addition to this model, there are several versions of trained USE models with different objectives, including multilingual, size/performance and question-answer systems.

So, in our approach USE receives a 100 tweets samples for each user. In this way the convolutional network will transform them into a vector of 512 dimensions, using the language model that we had already learned and updating with the new textual samples from *Twitter*.

## 4 Experimental Settings and Results

In this section, we describe the experiments carried out in order to obtain the author profile of Twitter users. First, we describe the evaluation corpus, then *baseline* results are presented, and finally results are shown using our proposed methodology. This baseline results are obtained by the combination of the different types of features (bag or words and N-grams), trained on two classification algorithms and several preprocessing variants (without emojis, without slangs, etc.). For all baseline experiments, the TF-IDF (mentioned above) weighting scheme is used.

### 4.1 Corpus Description

For training and evaluating our AP approach we used the corpus of PAN2017 competition [15], which was compiled from Twitter in Spanish. Gender and age information has been provided by the users themselves based on an online questionnaire. The corpus consists of 600 users of various nationalities: Mexican, Colombian, Peruvian, Argentine, Chilean, Venezuelan. 50% are male and the other 50% female.

<sup>1</sup> <https://tfhub.dev/google/universal-sentence-encoder-multilingual/1>.

Gender	Authors	Tweets
Male	2100	21000
Female	2100	21000
Total	4200	42000

## 4.2 Experimental Settings and Results

We performed several experiments considering bag of words and character N-grams as features. For each feature set we evaluated the impact of specific preprocessing strategies. The author profiling models obtained with the different settings were evaluated in terms of  $F-1$ , *precision*, *recall* and *accuracy*. Table 1 shows the results obtained with the logistic regression classification algorithm. The *Characteristics* column indicates whether the word bag (BoW) or character N-gram (N-char) is used, the *Dim* column indicates the amount of features extracted and therefore features vector dimensionality. Accuracy assessment measures (STD, the standard deviation of accuracy) are computed. The *Preprocessing* column indicates which strategy was followed in each experiment; in this case *NONE* indicates that no preprocessing was performed in that experiment, *without Emojis* indicates that the emojis were removed, as well as URL's, Hashtags, etc. It can be seen that the preprocessing strategy with which the best results are obtained is when user mentions are removed, which allows to infer that these are the ones that provide less information regarding the gender of the person who wrote the tweet.

**Table 1.** Results of experiments performed to predict gender using bag of words and logistic regression classifier.

Characteristics	Dim.	Accuracy	STD	F-1	Precision	Recall	Preprocessing
BoW	199114	0.6923	0.0287	0.6927	0.6947	0.6923	NONE
BoW	199114	0.6926	0.0289	0.6930	0.6951	0.6926	Without emoticons
BoW	186798	0.6909	0.0318	0.6913	0.6934	0.6909	Without hashtags
<b>BoW</b>	<b>166070</b>	<b>0.6976</b>	<b>0.0299</b>	<b>0.6979</b>	<b>0.6994</b>	<b>0.6976</b>	<b>Without mentions</b>
BoW	43089	0.6090	0.0680	0.6409	0.7273	0.6090	Without slangs
BoW	136731	0.6926	0.0289	0.6930	0.6951	0.6926	Without URLs
BoW	199226	0.6925	0.0288	0.6929	0.6950	0.6925	Without emojis
BoW	27137	0.6497	0.0533	0.6830	0.7836	0.6497	ALL

Table 2 presents results of the gender identification using character 3-gram as feature set and logistic regression classification algorithms. It is observed that the best results are also found when removing the mentions of users, however when slangs are removed the algorithm performance drops considerably.

**Table 2.** Results of experiments performed to predict gender using character N-grams and the logistic regression classifier.

Characteristics	Dim.	Accuracy	STD	F-1	Precision	Recall	Preprocessing
N-char	2550956	0.6833	0.0248	0.6837	0.6858	0.6833	NONE
N-char	2545770	0.6836	0.0243	0.6841	0.6862	0.6836	Without emoticons
<b>N-char</b>	<b>2309400</b>	<b>0.6874</b>	<b>0.0279</b>	<b>0.6878</b>	<b>0.6896</b>	<b>0.6874</b>	<b>Without mentions</b>
N-char	2470050	0.6811	0.0289	0.6815	0.6839	0.6811	Without hashtags
N-char	613817	0.5677	0.0537	0.6284	0.7685	0.5677	Without slangs
N-char	2324031	0.6817	0.0251	0.6821	0.6842	0.6817	Without emojis
N-char	1461457	0.6836	0.0243	0.6841	0.6862	0.6836	Without URLs
N-char	276872	0.6240	0.0454	0.6717	0.7990	0.6240	ALL

Table 3 presents the evaluation measures of accuracy, recall, precision and F-1 score obtained by the Support Vector Machine when trained on the BOW feature set. It can be seen that the best results are obtained by removing the mentions of users and the worst when the *slangs* are removed with a difference between them of approximately 10%.

**Table 3.** Results of the experiments performed to predict gender using bag of words and support vector machine classifier.

Characteristics	Dim.	Accuracy	STD	F-1	Precision	Recall	Preprocessing
BoW	199114	0.6926	0.0294	0.6933	0.6964	0.6926	NONE
BoW	199226	0.6925	0.0293	0.6931	0.6963	0.6925	Without emojis
BoW	186798	0.6933	0.0290	0.6939	0.6971	0.6933	Without hashtags
<b>BoW</b>	<b>166070</b>	<b>0.6981</b>	<b>0.0294</b>	<b>0.6986</b>	<b>0.7009</b>	<b>0.6981</b>	<b>Without mentions</b>
BoW	199114	0.6925	0.0296	0.6932	0.6963	0.6925	Without emoticons
BoW	43089	0.5939	0.0595	0.6292	0.7236	0.5939	Without slangs
BoW	136731	0.6925	0.0296	0.6932	0.6963	0.6925	Without URLs
BoW	276872	0.6219	0.0391	0.6753	0.8157	0.6219	ALL

Table 4 presents the results of the gender identification using character 3-grams and as a classification algorithm the Support Vector Machines. Likewise, it is observed that the best results are obtained by removing the mentions of users and the worst results when the *slangs* are removed. However, in the case of characters 3-gram, accuracy difference between the two is approximately 15%.

### 4.3 Experimental Settings and Results Using Transfer Learning

Table 5 presents results of gender identification using *Universal Sentence Encoder* (USE) to obtain 512-dimensional feature vectors for each user, that is, the 100 tweets are reduced to one 512-dimensional vector. The logistic regression is used



as classification algorithm. Table structure is the same as the previous ones and in this case dimensionality of the feature vector is always 512. We present the measures of accuracy, recall, precision and F-1 score. It is observed that the best results in terms of accuracy are obtained by removing the mentions and the worst by replacing the *slangs*.

**Table 4.** Results of the experiments performed to predict gender using character N-gram and the support vector machine classifier.

Characteristics	Dim.	Accuracy	STD	F-1	Precision	Recall	Preprocessing
N-char	2550956	0.6817	0.0269	0.6824	0.6858	0.6817	None
N-char	2545770	0.6822	0.0270	0.6830	0.6863	0.6822	Without emoticons
<b>N-char</b>	<b>2309400</b>	<b>0.6930</b>	<b>0.0296</b>	<b>0.6938</b>	<b>0.6975</b>	<b>0.6930</b>	<b>Without mentions</b>
N-char	2470050	0.6814	0.0261	0.6821	0.6856	0.6814	Without hashtags
N-char	2324031	0.6822	0.0270	0.6830	0.6863	0.6822	Without emojis
N-char	613817	0.5417	0.0314	0.6354	0.8286	0.5417	Without slangs
N-char	1461457	0.6822	0.0270	0.6830	0.6863	0.6822	Without URLs
N-char	27137	0.6517	0.0547	0.6812	0.7723	0.6517	ALL

**Table 5.** Results of experiments using transfer learning features with the logistic regression classifier to identify gender

Characteristics	Dim.	Accuracy	STD	F-1	Precision	Recall	Preprocessing
USE	512	0.6986	0.0222	0.6989	0.6854	0.6998	NONE
USE	512	0.6977	0.0263	0.6981	0.6808	0.7002	Without emojis
USE	512	0.6989	0.0213	0.6991	0.6854	0.7001	Without emoticons
USE	512	0.7041	0.0241	0.7042	0.6946	0.7036	Without hashtags
USE	512	<b>0.7156</b>	<b>0.0255</b>	<b>0.7158</b>	<b>0.6972</b>	<b>0.7198</b>	<b>Without mentions</b>
USE	512	0.6794	0.0459	0.6856	0.7745	0.6864	Without slangs
USE	512	0.7001	0.0265	0.7004	0.6895	0.7005	Without URLs
USE	512	0.6864	0.0489	0.6900	0.7598	0.7029	ALL

Table 6 presents results of gender identification using *Universal Sentence Encoder* (USE) to obtain 512-dimensional word vectors and support vector machine as classification algorithm. Evaluation measures of accuracy, recall, precision and F-1 score are presented. As with the previous classifier, it is observed that the best results in terms of accuracy are obtained by removing the mentions and the worst by replacing the *slangs*. Although the results are in accordance with those obtained with traditional characteristics in terms of better and worse preprocessing, we can observe that with Universal Sentence Encoder the difference between them does not exceed 3%.

**Table 6.** Results of experiments using transfer learning with the support vector machine classifier to identify gender

Characteristic	Dim.	Accuracy	STD	F-1	Precision	Recall	Preprocessing
USE	512	0.7068	0.0218	0.7072	0.6838	0.7124	NONE
USE	512	0.7037	0.0281	0.7043	0.6705	0.7137	Without emojis
USE	512	0.7061	0.0213	0.7064	0.6833	0.7115	Without emoticons
USE	512	0.7080	0.0211	0.7084	0.6856	0.7134	Without hashtags
USE	512	<b>0.7198</b>	<b>0.0267</b>	<b>0.7201</b>	<b>0.6951</b>	<b>0.7270</b>	<b>Without mentions</b>
USE	512	0.6955	0.0408	0.6974	0.7340	0.7207	Without slangs
USE	512	0.7009	0.0277	0.7010	0.6987	0.6974	Without URLs
USE	512	0.6992	0.0485	0.7020	0.7563	0.7201	ALL

## 5 Conclusions

In this paper, we introduced an approach to perform the gender identification of Twitter users using transfer learning. The transfer learning technique is useful when there is no much data for properly training machine learning algorithms. In this case, we had available a corpus of 4200 Twitter users, which is relatively low for training from scratch a deep learning model.

Our approach is based on the *Universal Sentence Encoder* model to obtain low dimensional vectors of documents (Users' tweets) and use them as features to perform author profiling. To evaluate the quality of the vectors (representing all the *tweets* of a user) obtained by USE, we used them as features for training two machine learning algorithm that generally obtain good results in author profiling [1]. With these experiments, we show that these vectors allow us to identify the author's gender with an accuracy of 71.98%, when the mentions to users are removed, with an SVM classifier for the PAN 2017 corpus. We can observe that this result is better than the obtained with the traditional approach for gender classification.

We consider that a possible extension of this work is to evaluate other transfer learning techniques, such as the *Universal Language Model Fine-tuning (ULM-Fit)* [8], which has achieved very good results in text classification problems.

## References

1. Aragón, M.E., López-Monroy, A.P.: Author profiling and aggressiveness detection in Spanish tweets: Mex-a3t 2018. In: IberEval@ SEPLN, pp. 134–139 (2018)
2. Caruana, R.: Multitask learning: a knowledge-based source of inductive bias. In: Proceedings of the Tenth International Conference on Machine Learning, pp. 41–48 (1993)
3. Cer, D., Yang, Y., et al.: Universal sentence encoder. arXiv preprint [arXiv:1803.11175](https://arxiv.org/abs/1803.11175) (2018)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)

5. Finn, C.: Learning to Lear with gradients. Ph.D. thesis, University of California, Berkeley (2018)
6. Gómez-Adorno, H.M.: Extracción de características de texto basada en grafos sintácticos integrados. Ph.D. thesis, Instituto Politécnico Nacional (2008)
7. Gusfield, D.: Algorithms on strings, trees and sequences: computer science and computational biology. Cambridge University Press (1997)
8. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint [arXiv:1801.06146](https://arxiv.org/abs/1801.06146) (2018)
9. Iyyer, M., Manjunatha, V., Boyd-Graber, J., Daumé III, H.: Deep unordered composition rivals syntactic methods for text classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 1681–1691 (2015)
10. Jurida, S.H., Džanić, M., Pavlović, T., Jahić, A., Hanić, J.: Netspeak: linguistic properties and aspects of online communication in postponed time. *J. Foreign Lang. Teach. Appl. Linguist.* **3**(1), 1–19 (2016)
11. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint [arXiv:1408.5882](https://arxiv.org/abs/1408.5882) (2014)
12. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint [arXiv:1802.05365](https://arxiv.org/abs/1802.05365) (2018)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pp. 3111–3119 (2013)
14. 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. 1532–1443 (2014)
15. 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, pp. 750–784 (2016)
16. Schmidhuber, J.: Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook. Ph.D. thesis, Technische Universität München (1987)
17. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: *Advances in Neural Information Processing Systems*, pp. 640–646 (1996)