Identification Of Bot Accounts In Twitter Using 2D CNNs On User-generated Contents

Notebook for PAN at CLEF 2019

Marco Polignano, Marco Giuseppe de Pinto, Pasquale Lops, and Giovanni Semeraro

University of Bari ALDO MORO
via E. Orabona 4, 70125, Bari, Italy
marco.polignano@uniba.it, marcogiuseppe.depinto@uniba.it, pasquale.lops@uniba.it,
giovanni.semeraro@uniba.it

Abstract The number of accounts that *autonomously* publish contents on the web is growing fast, and it is very common to encounter them, especially on social networks. They are mostly used to post ads, false information, and scams that a user might run into. Such an account is called *bot*, an abbreviation of robot (a.k.a. social bots, or sybil accounts). In order to support the end user in deciding where a social network post comes from, bot or a real user, it is essential to automatically identify these accounts accurately and notify the end user in time. In this work, we present a model of classification of social network accounts in *humans* or *bots* starting from a set of one hundred textual contents that the account has published, in particular on Twitter platform. When an account of a real user has been identified, we performed an additional step of classification to carry out its gender. The model was realized through a combination of convolutional and dense neural networks on textual data represented by word embedding vectors. Our architecture was trained and evaluated on the data made available by the PAN Bots and Gender Profiling challenge at CLEF 2019, which provided annotated data in both English and Spanish. Considered as the evaluation metric the accuracy of the system, we obtained a score of 0.9182 for the classification Bot vs. Humans, 0.7973 for Male vs. Female on the English language. Concerning the Spanish language, similar results were obtained. A score of 0.9156 for the classification Bot vs. Humans, 0.7417 for Male vs. Female, has been earned. We consider these results encouraging, and this allows us to propose our model as a good starting point for future researches about the topic when no other descriptive details about the account are available. In order to support future development and the replicability of results, the source code of the proposed model is available on the following GitHub repository: https://github.com/marcopoli/Identification-of-Twitter-bots-using-CNN

1 Introduction

A bot can be considered as an automatic system capable of creating contents on the web independently, i.e., without a human being involved. They are often used as link aggre-

gators or as repositories of copyrighted contents such as movies, songs or images. In the past, they have also been used to alter public opinion about famous people through the inclusion of highly targeted comments, especially during political elections [6]. In other cases, they have been used for manipulating the stock market or to spread fake theories. Each bot is different and it can complete different tasks. Some are more complex and produce contents very similar to those of a human being. Others simply retweet or answer simple questions. Bots that provide a legal and potentially useful service are also present on the web, but before using them, it is always necessary to be aware that you do not have a human being on the other side and that such data could be reused for other purposes. A recent analysis by the Pew Research Center [1] shows that automated systems generate two-thirds of the links shared on Twitter and their usage by real users is continuously growing due to their extreme efficiency and availability. If on the one hand, these systems provide the real user with a service of content sharing, on the other hand they represent a severe risk to the privacy of the user. It is clear that any interaction with them is stored and after a sufficient number of iterations is sufficient to profile the user in order to mislead him with fake advertising, fake websites, phishing, and other malicious activities.

The identification of bots on social networks is a challenge that, in recent years, has involved many researchers in creating an accurate classification system with real-time response times. A bot detection system can, therefore, help to maintain the stability of the network and ensure the safety of users. State of the art systems have relied on the use of numerous features of the user profile of the accused person such as the number of followers, the number of tweets and retweets, the length of the name, the age and so on. In our solution, we use only a set of one hundred tweets per user without knowing any information about the account from which they come. This situation can make the task of classification much more complex and utterly different from what is already present in the literature. This choice, which could be considered as limiting, presents a broader application scenario. In this way, it is possible to identify bots even in contexts where little information is available about the account, such as in possible situations of privacy, a topic that has become increasingly important in recent years. Our solution, therefore, wants to be able to work correctly with the least amount of information possible, trying to focus on what are the particularities of the writing style of an automatic system and a real user. In this regard, it may, therefore, be interesting to learn more about any differences in the style of writing used by men and women. Consequently, we wanted to deepen the theme, trying to carry out a further step of classification male vs. female in case the system was able to classify the account as human. This task, known in the literature as author profiling, has been a factor of scientific interest for many years, demonstrating how this information can be effectively identified simply by analyzing the content produced on social media.

The rest of the article is organized as follow: we start with an overview of the state of the art techniques used to address the classification task in Sec. 2. In Sec. 3 we describe and detail the model used in the challenge PAN Bots and Gender Profiling challenge at CLEF 2019 [19], while in Section 4, we discuss the experimental analysis carried out on the available training and validation dataset. In Sec. 5, conclusions and future

[1] https://www.pewinternet.org/2018/04/09/bots-in-the-twittersphere/

developments close the article inviting to download the available code of the proposed model in order to allow everyone to continue its implementation for future studies.

2 Related Work

The research area concerning authorship attribution to short texts is the one that well encloses the goal of our model: the classification of social media accounts into a bot or a human being using the user-generated textual contents. In this field of research, many classification strategies have been presented in the last years. One of the first systems of the interest of community was the one proposed by Lee [12]. It was based on the concept of *honeypots* defined as fake websites used as traps to identify the main characteristics of online spammers. The collected data were used at a later stage of classification through standard machine learning algorithms such as logistic regression and J48 to successfully detect them. Approaches based on the use of lexical features such as characters, words, and n-grams, part of speech (pos) tags, number of punctuations and more, have been proposed for numerous tasks of authorship attribution [23] as an example for author verification [8], plagiarism detection [4], author profiling or characterization [7]. However, approaches based on the manual definition of features engineering, as the previous, are often high time consuming and inaccurate. As a consequence, it has become always more common to identify approaches based on a representation of text in a vectorial space automatically learned from a neural network [14].

In [1], the author exposes the concept of word embedding that can be summarized as a "learned distributed feature vector to represent the similarity between words". This concept has been exploited by Mikolov [14] through word2vec, a tool for implementing work embeddings, but also by for Pennington [16] that proposed GloVe and by Bojanowski [2] who implemented its n-gram based vector space named FastText. The benefits of using these vectorial representations are about the property that every vector has in this space. In particular, two words that are semantically related as an example "sport" and "football" results very similarly in that space. This property supports the encoding of a sentence in a new structure able to preserve semantic relations among words. Approaches based on neural networks that use word embeddings as representations of sentences have proved to be very promising and effective for text classification tasks even in domains close to the authorship attribution [10]. In 2017 Shrestha [22] presents an authorship attribution model for short texts using convolutional neural networks (CNNs)[11]. The architecture uses n-grams as input, suitably transformed into semantic embeddings through a vector space of size 300 learned directly from the training data using the word2vec approach. The results described by Shrestha in her work show the good predictive ability of deep neural models, especially when a very large amount of data is available for the training phase. Strategies based on the use of recurrent neural networks (RNNs)[21] were used also by Kudugunta [9] for the task of tweet-level bot detection. The authors use GloVe for the encoding of the words constituting the Tweets so that an RNN followed by more dense networks enriched by a context vector could be applied to the data. Account-level Twitter bots account detection is, instead, performed by more classical machine learning strategies such as Logistic Re-

gression and Random Forest Classifiers. The excellent results obtained by the authors confirm again the usefulness of deep learning approaches for dealing with the task.

It is not difficult to find in literature examples of approaches that use neural networks even for the task of identifying the gender of the author of short texts, such as those published on social media. The overview about the multimodal gender identification challenge at PAN 2018 [20], well describe how the close totality of the submitted approaches are based on CNNs, RNNs and, more in general, a combination of neural networks. Following the wave of positive results obtained by such approaches based on neural networks, in this work, we combined the contribution of word embeddings with that of CNNs and Dense neural networks. In particular, unlike the works analyzed, we decided to use word embeddings pre-trained on English and Spanish and 2D CNNs able to work on the tensorial representation of content generated by each account. The particularity of our model lies in the idea to work with one single model directly on all the contents generated by the user considering them as a single learning example of our net. Moreover, we were able to use the same network architecture for both the two different classification tasks (bot vs. human, male vs. female) by varying only the training data obtaining by the way good results.

3 The Proposed Model

The model of classification of Twitter accounts in bot or human and later in male or female proposed in this work is based on a neural architecture that through convolutional layers and dense layers is able to capture distant relationships among words. In particular it focus not only on word relations in the same short text but also it is able to detect relations among word in different pieces of text. The key idea behind the proposed approach is that in order to exploit the stylistic information contained in a set of contents produced by a user, it is necessary to work on them at the same time. It is immediately clear that you can imagine the account of a user $u \in U$ as a set of contents $C = c_{u1}, c_{u2}, c_{u3}, ..., c_{un}$ aggregated in the form of a matrix. It is possible to place each sentence as a row of the matrix and to tokenize them into words for creating the grid structure. This allows us to draw a structured representation of the profile of the user and to work on them with approaches, as CNNs, that are more commonly used for data in a grid-like topology (i.e. images).

Fig. 1 describes very generally the architecture of the model designed for facing the bot identification task at user-level. In order to feed the network, we started with the encoding of each user-generated content in a list of word embedding vectors. Starting from a word embedding matrix $S \in IR^{e \times |V|}$, where e is the size of the word embedding vectors and $|V|$ the cardinality of the dictionary pre-trained, we encoded each of the first *50 words* w_{jk} that composed the j-th user tweet $t_{uj} \in T_u$ as word embedding vector. Tweets with a number of words higher than 50 have been truncated. Words not found in the vector space are transformed into word embeddings through a random vector selected from the entire dictionary, as proposed by Zhang [24]. We obtained a matrix of encoded user tweets $E_u \in IR^{m \times 50 \times e}$ where m is the number of tweets considered for the user, 50 is the fixed number of word of each tweet we decided to consider during the encoding, e is the size of the word embedding vectors. We used E_u as input of our

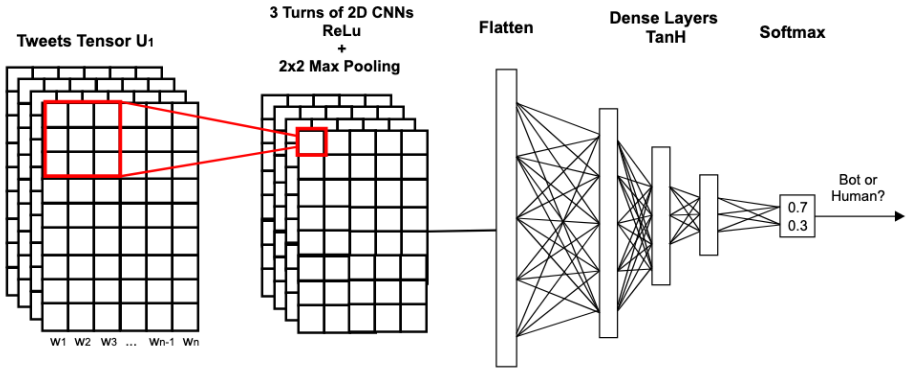

Figure 1. Overview of the Proposed Classification Model based on 2D CNNs and Dense Layers

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 96, 46, 200)	1500200
max_pooling2d_1 (MaxPooling2D)	(None, 48, 23, 200)	0
conv2d_2 (Conv2D)	(None, 44, 20, 100)	400100
max_pooling2d_2 (MaxPooling2D)	(None, 22, 10, 100)	0
conv2d_3 (Conv2D)	(None, 20, 8, 20)	18020
max_pooling2d_3 (MaxPooling2D)	(None, 10, 4, 20)	0
flatten_1 (Flatten)	(None, 800)	0
dense_1 (Dense)	(None, 400)	320400
dense_2 (Dense)	(None, 200)	80200
dense_3 (Dense)	(None, 100)	20100
dense_4 (Dense)	(None, 2)	202

Total params: 2,339,222
Trainable params: 2,339,222
Non-trainable params: 0

Figure 2. Detailed description of the architecture of the proposed classifier

architecture. Obviously, during the training phase, the matrix will already labeled with the class value. During the forecast phase, instead, the class will be properly estimated. The first three layers of our net are composed by 2D convolutional networks mediated by three max-pooling 2 x 2 operations. These max-pooling operations have the purpose of allowing the network to focus on low-level feature blocks and to map them in a higher-level feature space, more descriptive than the previous, through the convolutional operations. In this way, CNNs allow us to efficiently detect local relations shared among words closed in positions in user's tweets.

The details about the number of parameters for each layer and its shape are reported in Fig. 2. The figure allows us to observe the configuration of parameters used in particular for the first three CNNs and max-pooling operations. Starting from a word embedding matrix of size 100 x 50 x 300, we performed the first 2D CNN with a 5 x 5 kernel using a ReLu activation function [15]. The rectified linear unit (ReLu) can be formalized as $g(z) = max\{0, z\}$ and it allows to obtain a large value in case of activation by applying this function as a good choice to represent hidden units. As commonly performed, after a CNN able to perform convolutional operations on data, we applied a max-pooling operation of size 2 x 2 transforming each squared portion of our previous output as their max value. Pooling can, in this case, help the model to focus only on principal characteristics of every portion of data making them invariant by their position. We have run the process described twice more by simply changing the kernel size to 5 x 4 and finally to 3 x 3. Following we have flatten the output to make it compatible with the next layers.

We applied three Dense layers (fully connected layers) with a *tanh* activation function, varying the output size from 800 elements to 400, 200, and finally 100. This step allows the network to put in relations all the intermediate results obtained by the previous layers on the different sections of the input, discovering hidden correlations that can contribute to the prediction. The tanh function defined as $g(z) = 2\sigma(2z) - 1$ has an S-Shape and produces values among -1 and 1 making layer output more center to the 0. Moreover, it produces a gradient larger than sigmoid function helping to speed up the convergence [5]. Finally, another dense layer with a soft-max activation function has been applied for estimating the probability distribution of each of the classes.

4 Model Evaluation

The evaluation of the proposed model has been carried out in order to be able to answer two different research questions. First, we want to investigate whether the model produces results of accuracy that are good enough to identify Twitter bot accounts and to detect the gender of the user in case of human being. Secondly, we want to understand if the choice of word embedding vector space influences the final performance of the model. Such a result may produce interesting considerations to be used in future work on the subject by providing a detailed starting point.

4.1 Datasets, Baselines and Metrics

The dataset used for the training and validation phase of the model is the one provided by the PAN Bots and Gender Profiling challenge at CLEF 2019 [19]. It has been provided in two languages English and Spanish and it is composed by tweets labeled with a bot or to a male/female human class. The detailed statistics about the dataset are available in Tab. 1 . The test dataset has been not released yet.

We used as validation metric the *Accuracy*, the same used in the PAN challenge. In particular, for each language, individual accuracies are calculated. Firstly, we calculated the accuracy of identifying bots vs. human. Then, in case of humans, we calculated the

Table 1. PAN 2019 - Bot and Gender Profiling task dataset

	English		*Spanish*	
	#Accounts	**#Tweets**	**#Accounts**	**#Tweets**
Bot	2060	206000	1500	150000
Human	2060	206000	1500	150000
Male	1030	103000	750	75000
Female	1030	103000	750	75000

accuracy of identifying males vs. females. Finally, the average of the accuracy values per language are used for obtain the final score.

The baseline considered for our task is the majority vote classifier that, as consequence of the homogeneous classes distribution in the dataset has an accuracy score equal to 0.5.

4.2 Tweets Preprocessing and Data Enrichment

The tweets available in the dataset were supplied without any cleaning or pre-processing. In this regard, it was decided to keep as much information as possible within the model. Each sentence is divided into tokens through the TweetTokenizer class of the NLTK library. After that, we used Ekphrasis preprocessor library to annotate mentions, URL, email, numbers, dates, amount of money, and make word spelling correction and hashtag unpacking if necessary. As an example "14/01/2018" is translated into the word "date" and the hashtag "#lovefootball" is translated into "love football". This process allows to correctly apply the translation of words in word embedding without excluding the previously mentioned elements.

In order to obtain a more general and accurate model, we have decided to enrich the reference dataset with further data regarding the contents produced by men or women. We followed the strategy used in the GitHub project Gender-Classification-using-Twitter-Feeds [2] based on the work of Liu [13]. For the English language, we developed a module able to collect tweets localized in the USA through official Twitter API. Messages collected between the 15/03/2019 to the 30/04/2019 have been filtered on the base of the official first name written in the account description. In particular, if the name was one of them reported by the USA census [3] as used for a male subject we labeled it as a consequence. Similarly, we labeled female tweets[4]. As far as Spanish is concerned, the strategy was symmetrical. First of all, we collected the geolocalized tweets in Spain, then we filtered and categorized them following a list of male and female names available on the web[5].

The previously filtered tweets were then aggregated into groups of 100, randomly selecting them from the corresponding reference class. We obtained 1000 additional examples of annotated accounts for each of the classes and for each of the two languages.

[2] https://github.com/vjonnala/Gender-Classification-using-Twitter-Feeds

[3] http://www2.census.gov/topics/genealogy/1990surnames/dist.male.first

[4] http://www2.census.gov/topics/genealogy/1990surnames/dist.female.first

[5] http://www.20000-names.com/male_spanish_names.htm

4.3 Word Embedding Pre-trained Vectors

Word embeddings could be trained directly on "training data" of the domain of application, but this strategy can lack generalization. When a new sentence to classify is provided as input many words in it could be not possible to be translated making impossible the correct classification. For this reason, we decided to use a common practice of transfer learning in NLP tasks i.e. the use of vector spaces word embeddings already pre-calculated on different domains. This allows us to cover an extensive variety of terms by reducing the computational cost of the model and including information about terms that are independent of their domain of use. We decided to compare the results of the model obtained varying three different pre-trained word embeddings:

- **Google word embeddings (GoogleEmb)** [6]: 300 dimensionality word2vec vectors, case sensitive, composed by a vocabulary of 3 million words and phrases that are obtained from roughly 100 billion of tokens extracted by a huge dataset of Google News;
- **GloVe (GloVeEmb)**[7]: 300 dimensionality vectors, composed by a vocabulary of 2.2 million words case sensitive obtained from 840 billion of tokens and trained on data crawled from generic Internet web pages;
- **FastText (FastTextEmb)**[8]: 300 dimensionality vectors, composed by a vocabulary of 2 million words and n-grams of the words, case sensitive and obtained from 600 billion of tokens trained on data crawled from generic Internet web pages by Common Crawl nonprofit organization;

4.4 Experimental Runs And Results

The model has been trained using the categorical cross entropy loss function [5] and Adam optimizer for 20 epochs and best models have been used for the classification phase. The number of batches is set as 64 for optimization reason.

In order to evaluate the influence that the different pre-trained word embeddings have on the final performances of the model, we have performed the training phase keeping constant all the parameters except the word embedding matrix used to encode the user's tweets. The portion of the dataset used for the evaluation is equal to 20% of the training set using 42 as a random seed. As a consequence of the not availability of GoogleEmb and GloVe pre-trained vectors for the Spanish language, we evaluate the model performances only on English data.

The results in Tab. 2 showed quite better performances obtained by FastText word embedding vector space. The differences among results are not statistically significant but this allow us to decide to use FastText in our final model. Moreover, its availability in both the languages of the interest of the PAN Bots and Gender Profiling challenge at CLEF 2019 [19] (English and Spanish) has emphasized and confirmed our choice.

The final model trained four times (Eng - Bot vs. Human, Eng - Male vs. Female, Esp - Bot vs. Human, Esp - Male vs. Female) has been deployed on Tira.io [9] [17]

[6] https://goo.gl/zQFRx3
[7] https://nlp.stanford.edu/projects/glove/
[8] https://fasttext.cc/docs/en/english-vectors.html
[9] https://www.tira.io

Table 2. Results of the model on the validation set varying the word embedding encoding

	English	
	Bot vs. Human	Male vs. Female
Baseline	0.5000	0.5000
GoogleEmb	0.9547	0.8173
GloVe	0.9611	0.8264
FastText	***0.9733***	***0.8479***

Table 3. Results obtained on the test set considering organizers baselines

	English		Spanish	
	Bot vs. Human	Male vs. Female	Bot vs. Human	Male vs. Female
Our Model	0.9182	0.7973	**0.9156**	**0.7417**
char nGrams	**0.9360**	0.7920	0.8972	0.7289
word nGrams	0.9356	**0.7989**	0.8833	0.7244
W2V	0.9030	0.7879	0.8444	0.7156
LDSE	0.9054	0.7800	0.8372	0.6900

for participating to the PAN 2019 Bots and Gender Profiling challenge. The first run on a preliminary released test set showed an accuracy score of 0.9470 and 0.8181 respectively for the Bot vs. Human and Male vs. Female tasks in English language. In Spanish language we obtained 0.9611 and 0.7778 respectively for the Bot vs. Human and Male vs. Female tasks. Finally we run the model also on the final test set earning an accuracy score of **0.9182** and **0.7973** respectively for the Bot vs. Human and Male vs. Female tasks in **English language**. In **Spanish language** we obtained **0.9156** and **0.7417** respectively for the Bot vs. Human and Male vs. Female tasks.

A comparison with the baselines proposed by the challenge authors after the end of the competition [19] is showed in Fig. 3. In particular, it is interesting to observe that our model is better than all the proposed baselines for the Spanish language. For English, a simple strategy based on n-grams and Random Forest can overcome our results. Probably this anomaly has been obtained as a consequence of the ability of n-grams to capture words misspelled or unknown in a standard vocabulary like the one used in FastText embedding space. In any case, it is performed more better than the LDSE baseline [18]. A possible future work can try to overcome these limits using one of the newest language models, as BERT [3], as a base for our proposed network.

5 Conclusion

In this work, we presented an architecture based on deep neural networks for the classification of Twitter accounts in a bot or human, carrying out, where possible, a further refinement in man or woman. The model is based on the representation of the user account as a list of 100 tweets appropriately transformed into tensorial form through the encoding of words in the equivalent word embedding FastText form. Unlike state of the art, no additional information about the Twitter account has been used allowing our

approach to be used even when this information is not available. On these representations of input data, we learned a deep neural network that uses 2D CNNs, max-pooling operations, and Dense Layers to estimate the probability of distribution of annotation classes correctly. The approach was tested on the data provided by the PAN Bots and Gender Profiling challenge at CLEF 2019, which provided appropriately annotated data in both English and Spanish. As a preliminary step, the model was validated by varying the pre-trained word embedding space used in the training phase. The embeddings provided by Google and trained on News, GloVe trained on Tweets and FastText trained on web pages collected on the web were tested. The final choice fell on FastText following the good results obtained and its effectiveness in the tasks of classification of the text demonstrated in the literature. The final model learned was submitted to the competition and obtained a score of accuracy equal to: 0.9182 and 0.797 respectively for the Bot vs. Human and Male vs. Female tasks in English language, 0.9156 and 0.7417 for Spanish data.

The good results obtained suggest that such approaches based on deep neural networks are an excellent basis for solving the task. In particular, they are general enough to work in a real application context correctly. Since the result obtained can only be considered a starting point for further extensions, modifications, and improvements of the proposed approach, we have decided to make the product code available to the whole reference community with the hope that it will be useful for future work in the field. The code that implements the presented model can be found at the following GitHub repository: https://github.com/marcopoli/Identification-of-Twitter-bots-using-CNN

6 Acknowledgment

This work is partially funded by project "Electronic Shopping & Home delivery of Edible goods with Low environmental Footprint" (ESHELF), under the Apulian INNONETWORK programme, Italy. Moreover, it is partially funded by project "DECiSION" codice raggruppamento: BQS5153, under the Apulian INNONETWORK programme, Italy.

References

1. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research 3(Feb), 1137–1155 (2003)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the ACL 5, 135–146 (2017)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. zu Eissen, S.M., Stein, B., Kulig, M.: Plagiarism detection without reference collections. In: Advances in Data Analysis, Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation e.V., Freie Universität Berlin, March 8-10, 2006, pp. 359–366 (2006), https://doi.org/10.1007/978-3-540-70981-7_40
5. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)

6. Howard, P.N., Woolley, S., Calo, R.: Algorithms, bots, and political communication in the us 2016 election: The challenge of automated political communication for election law and administration. Journal of information technology & politics 15(2), 81–93 (2018)
7. Koppel, M., Argamon, S., Shimoni, A.R.: Automatically categorizing written texts by author gender. Literary and linguistic computing 17(4), 401–412 (2002)
8. Koppel, M., Schler, J.: Authorship verification as a one-class classification problem. In: Proc. of the twenty-first international conference on Machine learning. p. 62. ACM (2004)
9. Kudugunta, S., Ferrara, E.: Deep neural networks for bot detection. Information Sciences 467, 312–322 (2018)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature 521(7553), 436 (2015)
11. LeCun, Y., et al.: Generalization and network design strategies. Connectionism in perspective pp. 143–155 (1989)
12. Lee, K., Caverlee, J., Webb, S.: Uncovering social spammers: social honeypots+ machine learning. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 435–442. ACM (2010)
13. Liu, W., Ruths, D.: What's in a name? using first names as features for gender inference in twitter. In: 2013 AAAI Spring Symposium Series (2013)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
15. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
16. 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)
17. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF. Springer (2019)
18. Rangel, F., Franco-Salvador, M., Rosso, P.: A low dimensionality representation for language variety identification. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 156–169. Springer (2016)
19. Rangel, F., Rosso, P.: Overview of the 7th author profiling task at pan 2019: Bots and gender profiling. In: Cappellato L., Ferro N., MÃijller H, Losada D. (Eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org (2019)
20. 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. Working Notes Papers of the CLEF (2018)
21. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science (1985)
22. Shrestha, P., Sierra, S., Gonzalez, F., Montes, M., Rosso, P., Solorio, T.: Convolutional neural networks for authorship attribution of short texts. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 669–674 (2017)
23. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology 60(3), 538–556 (2009)
24. Zhang, Z., Robinson, D., Tepper, J.: Detecting hate speech on twitter using a convolution-gru based deep neural network. In: European Semantic Web Conference. pp. 745–760. Springer (2018)