A Hierarchical Neural Network Approach for Bots and Gender Profiling

Notebook for PAN at CLEF 2019

Andrea Cimino and Felice dell'Orletta

Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC–CNR)
{andrea.cimino,felice.dellorletta}@ilc.cnr.it

Abstract In this paper we describe our participation in the Bots and Gender Profiling shared task of PAN@CLEF2019 for the English language. We tested three approaches based on three different document classification algorithms. The first approach is based on a SVM classifier with handcrafted features using a wide set of linguistic information. The second and the third approaches exploit recent advances in Natural Language Processing using a Hierarchical GRU-LSTM Neural Network using word embeddings trained on Twitter and finally an adaptation of the BERT system. After an in-house evaluation, we submitted the final run with the Hierarchical Neural Network model, which achieved a final accuracy of 0.9083 in the Bots Profiling task and a score of 0.7898 in the Gender Profiling task.

1 Introduction

Nowadays the increased importance of social media platforms in everyday life has made the users of these platforms extremely impressionable by messages and posts written by companies, political parties or even social media influencers. In the recent years it has been shown that such platforms were exploited in order to diffuse fake news or for commercial activities using sophisticated techniques, such as very smart Bots, for massive mind manipulation. For this reason, the biggest social media platforms such as Facebook or Twitter started using algorithms to automatically detect and delete such Bot accounts, but latest advances in Natural Language Generation such as GPT-2 [1], makes the automatic Bot detection still a challenging problem. One of the approaches commonly used by these platforms in order to detect a Bot, is the classification of a set of documents (e.g. tweets) rather than a single document, since usually the set of documents written by a Bot follows a common lexical and stylistic pattern. With respect to the previous PAN shared task, the PAN 2019: Bot and Gender profiling task [2] introduces the novelty of asking to participants to identify, given a set of tweets, the type of a user (Bot or human) and, in case the type of user is classified as human, to predict the gender (male or female). We addressed the Bots and genre profiling task as

a 3–class classification problem and we developed three different classifier models: one that uses a classic approach based on the extraction of linguistic features and lexicons lookups using the linear SVM algorithm, and two more recent neural network based solutions. The first is based on a hierarchical GRU-LSTM deep neural network, and the second on a language model based neural network (BERT), which we adapted in order to handle large documents.

This paper makes the following contributions:

1. We propose a comparison between a more classical classification approach with extremely handcrafted features learned by a linear SVM algorithm and a low-engineered approach based on neural network models.
2. We show that the Hierarchical GRU-LSTM deep neural network has better performance with respect BERT, a very famous pretrained language model based on neural network.

In the following sections of this paper, we first explain the related work in Section 2. The preprocessing step and the external resources used are described in Section 3. Our models are properly described in Section 4. The details of the experiments used to confirm the model performances are described in Section 5. Finally, we conclude the paper and outline future work in Section 6.

2 Related works

This year edition of the PAN Bots and Gender Profiling shared task focuses on automatic identification on the Twitter platform of the type of user (BOT or human), and in case of a human, to detect the human gender. This is a slightly different version of previous year shared task [3], where participants where asked to identify the gender based not only on textual content, but exploiting also images posted on the Twitter platform. For what concerns the classification task using only the textual component, the linear SVM learning algorithm with heavily engineered features has been shown to be very effective method for identification of the gender. Daneshvar and Inkpen [4] used word n–grams and character n–grams with dimensionality reduction techniques and achieved the best score on the English and Spanish language. A similar solution was used by the second best participant (Tellez et al. [5]) that achieved the second best score on the English language and the best score on the Arabic language. Surprisingly, deep learning based sequential models did not achieve very good results when just considering the textual components. The best model on the English language that used this kind of architecture was presented by Takashi et al. [6]. The authors used a textual component composed by a word embedding, recurrent neural network (GRU), pooling, and fully connected layers. When tested on the English language, they achieved an accuracy of 0.7864, 4 points less than the state of the art. On the other hand their model, when mixed with visual information, achieved the average best scores among the Arabic, Spanish and English language, showing that deep learning architectures have strong results specially when combining multi–modal information.

3 Preprocessing and resources

The training dataset provided by the task organizers consists of 4,120 examples, each example containing a set of tweets which were written on the Twitter platform by a bot, a male or a female. In our approach we concatenated the tweets contained in each sample to produce the document, which is our classification unit. We used the "SEP" token as tweets separator in order to preserve the tweet length information which is used by our models. Since the SVM model relies on morpho-syntactically tagged texts, both training and test data were automatically morpho-syntactically tagged by our POS tagger described in [7]. In addition, in order to improve the overall accuracy of our models, we used an existing sentiment polarity lexicon and developed a word embedding lexicon for English tweets.

3.1 Sentiment Polarity Lexicon

We used the SentiWordnet 3.0 sentiment polarity lexicon [8]. This is a freely available lexicon for the English language[1] and includes more than 117,000 English word entries. It was automatically created using a semi–supervised step and a final random-walk step for refining the final positive and negative polarity scores.

3.2 Word embedding Lexicon

In order to extract semantic information from words we created a word embedding lexicon using the word2vec[2] toolkit [9]. As recommended in [9], we used the CBOW model that learns to predict the word in the middle of a symmetric window based on the sum of the vector representations of the words in the window. For our experiments, we considered a context window of 5 words. These models learn lower-dimensional word embeddings. The word embedding lexicon was built using a set of 19,700,117 English tweets downloaded from the Twitter platform. In order test the contribution of the embeddings in classification w.r.t. the vector size, we generated 16, 32, 64 and 128 sized vectors.

4 The proposed models

In this section we will describe the 3 devised models proposed for our participation in the Bots and Gender profile shared task.

4.1 The SVM Model

The SVM classifier exploits a wide set of features ranging across different levels of linguistic description. All these features were already tested in our previous participation at the EVALITA 2018 [10], the periodic evaluation campaign of Natural Language Processing (NLP) and speech tools for the Italian language.

[1] https://github.com/aesuli/sentiwordnet
[2] http://code.google.com/p/word2vec/

The features are organised into three main categories: *raw and lexical text features*, *morpho-syntactic features* and *lexicon features*. All the calculated features are the input of the linear SVM algorithm implemented in the liblinear [11] library which finally generates the final statistical model used then to classify unseen documents.

Raw and Lexical Text Features
Number of tokens: The average number of tokens of an analyzed tweet.
Character n-grams: presence or absence of contiguous sequences of characters in the analyzed tweets.
Word n-grams: presence or absence of contiguous sequences of tokens in the analyzed tweets.
Lemma n-grams: presence or absence of contiguous sequences of lemma occurring in the analyzed tweets.
Repetition of n-grams chars: presence or absence of contiguous repetition of characters in the analyzed tweets.
Number of mentions: number of mentions (@) occurring in the analyzed tweets.
Number of hashtags: number of hashtags occurring in the analyzed tweets.
Punctuation: the number of tweets that ends with one of the following punctuation characters: "?", "!".

Morpho-syntactic Features
Coarse grained Part-Of-Speech n-grams: presence or absence of contiguous sequences of coarse–grained PoS, corresponding to the main grammatical categories (noun, verb, adjective).
Fine grained Part-Of-Speech n-grams: presence or absence of contiguous sequences of fine-grained PoS, which represent subdivisions of the coarse-grained tags (e.g. the class of nouns is subdivided into proper vs common nouns, verbs into main verbs, gerund forms, past particles).
Coarse grained Part-Of-Speech distribution: the distribution of nouns, adjectives, adverbs, numbers in the tweets.

Lexicon features
Emoticons: presence or absence of positive or negative emoticons in the analyzed tweet. The lexicon of emoticons was extracted from the site *http://it.wikipedia.org/wiki/Emoticon* and manually classified.
Lemma sentiment polarity n-grams: for each n-gram of lemmas extracted from the analyzed tweet, the feature checks the polarity of each component lemma in the existing sentiment polarity lexicons. Lemma that are not present are marked with the *ABSENT* tag. This is for example the case of the trigram *all very nice* that is marked as "*ABSENT-POS-POS*" since *very* and *nice* are marked as positive in the considered polarity lexicon and *all* is absent. The feature is computed exploiting the SentiWordnet 3.0 lexicon resource.
Polarity modifier: for each lemma in the tweets occurring in the existing sentiment polarity lexicons, the feature checks the presence of adjectives or adverbs in a left context window of size 2. If this is the case, the polarity of the lemma is assigned to the modifier. This is for example the case of the bigram *not interesting*, where "interesting" is

a positive word, and “not” is an adverb. Accordingly, the feature “not_POS” is created. The feature is computed exploiting the SentiWordnet 3.0 lexicon resource.
Distribution of sentiment polarity: this feature computes the percentage of positive, negative and neutral lemmas that occur in the tweets. To overcome the sparsity problem, the percentages are rounded to the nearest multiple of 5. The feature is computed exploiting the SentiWordnet 3.0 lexicon resource.
Most frequent sentiment polarity: the feature returns the most frequent sentiment polarity of the lemmas in the analyzed tweets. The feature is computed exploiting the SentiWordnet 3.0 lexicon resource.
Word embeddings combination: the feature returns the vectors obtained by computing separately the average of the word embeddings of the nouns, adjectives and verbs of the tweet, obtaining a total of 3 vectors for each tweet. If a specific morphosyntactic category is not present, a feature indicating the absence of such category is added.

4.2 The BERT Model

Following the latest advances in NLP, we wanted to test how well pretrained language model representations behave on the Bot and Gender stylistic profiling shared task. Context-free models such as word2vec generate a single vector for each word, which is independent by the context in which the word is found. For example, the word "bank" can have two different meanings with respect to the context in which the word is surrounded. Language models like BERT [12] or ELMo [13] allow to obtain a distinct vector for each word based also on the context, which make such models very suitable for many NLP downstream tasks. In order to test the performance of these models, we choose BERT since Google provides pretrained models [3], which need only to be fine-tuned with an inexpensive procedure. Among the models available on the github repository, we choose the recommended model: BERT-Base Multilingual Cased which is trained on 104 languages with 110M parameters. One of the limitations of this pretrained model is that such model was trained on sentences not longer than 512 tokens, which made the standard fine-tuning procedure not suitable for our case, since the training documents (the concatenation of the tweets) were much longer than 512 tokens. For this reason, we generated 5 different fined tuned downstream tasks models by considering 5 chunks of 500 tokens each. In testing phase, each document was still divided in 5 chunks. Each chunk was then classified by the previously 5 fine tuned models. We then choose as winning class among BOT, male and female, the majority class resulting by all the predictions of the 5 models on the 5 chunks.

4.3 The Hierarchical GRU/LSTM Model

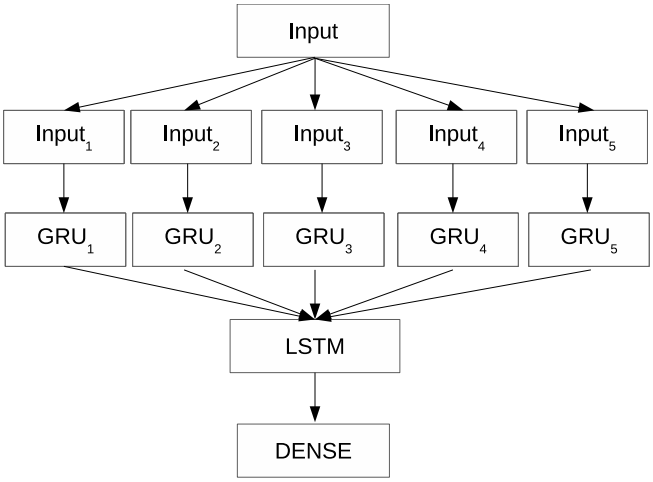
GRU units are able to propagate an important features that came early in the input sequence over a long distance, thus capturing potential long-distance dependencies. Unfortunately, it has been shown that long dependencies are lost in case of very long sequences. For this reason, since we treat the batch of tweets to be classified as single

[3] `https://github.com/google-research/bert`

document, we resorted to a two-layer hierarchical GRU/LSTM architecture. In addition, each document containing the set of tweets to be analyzed is first truncated to the first 2500 tokens. This operation is done since in-house experiments have not shown a significant drop in performance w.r.t. analyzing all the tweets contained in a single example. Moreover, the truncation allows a faster training in terms of time, considering the number of tweets belonging to the training set. Each document is then split in 5 chunks of 500 tokens, which are the input of five different GRU unit (48 dimensions), which produce 5 "chunk" embeddings. Finally, all the chunk embeddings are the input of a final LSTM (48 dimensions) layer. Figure 1 shows a graphical representation of the hierarchical GRU-LSTM architecture. We applied a dropout factor to both input gates and to the recurrent connections in order to prevent overfitting which is a typical issue in neural networks [14]. We have chosen a dropout factor value of 0.55. For what concerns the optimization process, the categorical cross entropy function is used as a loss function and optimization is performed by the rmsprop optimizer [15].

Furthermore, we performed a 5-fold training approach. More precisely we build 5 different models using different training and validation sets. These models are then exploited in the classification phase: the assigned labels are the ones that obtain the majority among all the models. The 5-fold approach strategy was chosen in order to generate a global model which should less be prone to overfitting or underfitting w.r.t. a single learned model.

Figure 1: The Hierarchical GRU-LSTM architecture



Each input word is represented by a vector which is composed by:
Word embeddings: the word embedding extracted by the available word embedding lexicon (32 dimensions), and for each word embedding an extra component was added to handle the "unknown word" (1 dimension).

Word polarity: the corresponding word sentiment polarities obtained by using the SentiWordnet 3.0 resource. This results in 3 components: 2 used for positive and negative values found in the resource, and one binary component set to 1 in case the word is not found in the lexicon.
Is capitalized word: a component (1 dimension) indicating whether the word is capitalized.
Is uppercased word: a component (1 dimension) indicating whether the word is uppercased.
Is URL: a component (1 dimension) indicating whether the word is an URL.
Is hashtag: a component (1 dimension) indicating whether the word is an hastag.
Is mention: a component (1 dimension) indicating whether the word contains a mention.
Is separator: a component (1 dimension) indicating if the word is the "SEP" reserved token, which we use to divide the tweets.

5 Experiments and Results

In order to choose the model to submit for the final run of the Bots and Gender profiling shared task, we tested all the 3 devised models on the official development set distributed by the organizers. The development set was composed by 1,240 examples: 640 composed by BOT messages, 310 by males and 310 by females. The training data (including the development set) is composed by 4,120 examples.

Configuration	Bot F-score	Male F-score	Female F-score	Avg F-score
linear SVM	0.94	0.71	0.59	0.746
Hierarchical GRU/LSTM 5 Fold	**0.92**	**0.76**	0.74	**0.806**
BERT Multi	0.90	0.72	**0.71**	0.776

Table 1: Classification results of the proposed models on the official development set.

Table 1 reports the overall accuracies achieved by our proposed models on the official developments set. Is it worth noting that, since an official software evaluator was not distributed among with the training data, we developed our own internal evaluator. It can be noticed that the proposed models behave quite well in average: the average f-score ranges between 0.74 for the linear SVM to 0.80 for the hierachical GRU/LSTM model. Surprisingly, the Hierarchical GRU/LSTM model outperformed the BERT system. We think that such difference in performance is due to many reasons. First of all, the BERT pretrained language model is trained on generic texts and not on social media texts. Another possible cause of the difference in terms of performance w.r.t. to the GRU/LSTM model is that BERT does not use any handcrafted feature w.r.t. the other two models (SVM in particular). As it is shown in Table 1, these handcrafted features could be the the reason of the highest performance of the SVM model when considering the BOT vs human classification task.

The obtained results on the development set lead us to choose the Hierarchical GRU/LSTM model for the final runs since this model behaves better when considering the average f–score on the 3 tasks (0.806). Table 2 reports the results obtained on the official test set. The result was obtained using the official scorer provided by the organizers on the TIRA[16] evaluation platform. In addition the table reports the baselines provided by the shared task organizers which are char n–grams, word n–grams, word2vec and LDSE [17] based and which are fully described in [2].

Dataset	**Bot vs Human**	**Male vs Female**
GRU-LSTM model	0.9083	0.7898
char nGrams baseline	0.9360	0.7920
word nGrams baseline	0.9356	0.7989
W2V baseline	0.9030	0.7879
LDSE baseline	0.9054	0.7800

Table 2: Classification results of Hierarchical GRU/LSTM model and of the baselines provided by the shared task organizers on the official test set.

For what concerns the Bot vs Human task, we can notice that our proposed model outperformed both the W2V and LDSE baselines, but char n-grams and word-ngrams bases lines performed better than our model (+3% in accuracy). This suggest that these features are very important for this classification task. Such behaviour was shown also in our internal tests, but the gain in terms of accuracy was less than what was shown in the test set (+2% in accuracy).

For what concerns the Male vs Female task, also here our GRU-LSTM model performed well, being in line with all the proposed baselines. Unfortunately all the errors in classification made in the in the Bot vs Human task were propagated in the Male vs Female task. So most probably a combination of a SVM based model for the Bot vs Human task and the GRU-LSTM model for the Male vs Female task would result in the best solution to achieve the best scores.

6 Conclusion

We presented three systems for the Bots and Gender Profiling shared task, on a SVM classifier with handcrafted features using a wide set of linguistic information. The second and the third based on Hierarchical GRU-LSTM on on the the BERT system. After internal experiments, we participated with the Hierarchical GRU-LSTM model, which showed promising results, outperforming all the W2V and LDSE baselines. It would be interesting to incorporate char-level features in our GRU-LSTM in order to evaluate the difference in terms of performance w.r.t. our current model, which is only token based at the moment.

References

1. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. OpenAI Blog. 2019.
2. Francisco Rangel, Paolo Rosso. Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In: Cappellato L., Ferro N., Müller H, Losada D. (Eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org
3. Francisco M. Rangel Pardo, Paolo Rosso, Manuel Montes-y-Gómez, Martin Potthast, Benno Stein. Overview of the 6th Author Profiling Task at PAN 2018: Multimodal Gender Identification in Twitter. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. CEUR Workshop Proceedings. CEUR-WS.org.
4. Saman Daneshvar and Diana Inkpen. Gender Identification in Twitter using N-grams and LSA: Notebook for PAN at CLEF 2018. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. CEUR Workshop Proceedings. CEUR-WS.org.
5. Eric Sadit Tellez, Sabino Miranda-Jiménez, Daniela Moctezuma, Mario Graff, Vladimir Salgado and José Ortiz-Bejar. Gender Identification through Multi-modal Tweet Analysis using MicroTC and Bag of Visual Words: Notebook for PAN at CLEF 2018. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. CEUR Workshop Proceedings. CEUR-WS.org.
6. Takumi Takahashi, Takuji Tahara, Koki Nagatani, Yasuhide Miura , Tomoki Taniguchi and Tomoko Ohkuma. Text and Image Synergy with Feature Cross Technique for Gender Identification: Notebook for PAN at CLEF 2018. In Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. CEUR Workshop Proceedings. CEUR-WS.org.
7. Andrea Cimino and Felice dell'Orletta. Building the state-of-the-art in POS tagging of Italian Tweets. In Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016.
8. Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Proceedings of the International Conference on Language Resource and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta
9. Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv1:1301.3781.
10. Andrea Cimino , Lorenzo De Mattei, Felice Dell'Orletta. Multi-task Learning in Deep Neural Networks at EVALITA 2018. Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018.
11. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research. Volume 9, 1871–187, 2008.
12. Jacob Devlin Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv1:1810.04805. http://arxiv.org/abs/1810.04805.
13. Matthew E. Peter, Mark Neumann. Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:

Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers).
14. Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. arXiv preprint arXiv:1512.05287
15. Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-RmsProp: Divide the gradient by a running average of its recent magnitude. In COURSERA: Neural Networks for Machine Learning. 2012.
16. Martin Potthast, Tim Gollub, Matti Wiegmann and Benno Stein. TIRA Integrated Research Architecture. Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF, Springer. 2019.
17. Francisco M. Rangel Pardo, Marc Franco-Salvador and Paolo Rosso. A Low Dimensionality Representation for Language Variety Identification. In Computational Linguistics and Intelligent Text Processing - 17th International Conference, CICLing 2016, Konya, Turkey, April 3-9, 2016, Revised Selected Papers, Part II.