

Contextualized BERT Sentence Embeddings for Author Profiling: The Cost of Performances

Marco Polignano(✉), Marco de Gemmis, and Giovanni Semeraro

Department of Computer Science, University of Bari Aldo Moro,
Via E. Orabona 4, 70125 Bari, Italy
{marco.polignano,marco.degemmis,giovanni.semeraro}@uniba.it

Abstract. The necessity to know information about the real identity of an online subject is a highly relevant issue in User Profiling, especially for analysis from digital sources such as social media. The digital identity of a user does not always present explicit data about her offline life such as age, gender, work, and more. This problem makes the task of user profiling complex and incomplete. For many years this issue has received a considerable amount of attention from the whole community, which has developed several solutions, also based on machine learning, to estimate user characteristics. The increasing diffusion of deep learning approaches has allowed, on the one hand, to obtain a considerable increase in predictive performance, but on the other hand, to have available models that cannot be interpreted and that require very high computational power. Considering the validity of new pre-trained language models on extensive data for resolving many natural language processing and classification tasks, we decided to propose a BERT-based approach (BERT-DNN) also for the author profiling task. In a first analysis, we compared the results obtained by our model with them of more classical approaches. As a follow, a critical analysis was carried out. We analyze the advantages and disadvantages of these approaches also in terms of resources needed to run them. The results obtained by our model are encouraging in terms of reliability but very disappointing if we consider the computational power required for running it.

Keywords: Language model · Author profiling · Classification · Machine learning · Deep learning · BERT

1 Introduction and Motivation

Social media plays a fundamental role in everyone's life nowadays. Our digital identity is vast and rich in details due to the many online platforms used to share information with other users about interests, places visited, or, more generally, what we are doing. This interesting data source has been extensively used to perform different types of analysis [15], such as sentiment analysis [19,25,26],

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12252, pp. 135–149, 2020.
https://doi.org/10.1007/978-3-030-58811-3_10

preference extraction [38], and holistic user profiling in general [18]. As a result of the increasingly stringent laws regarding the privacy of users GDPR [35], it is always less possible to access and use the data provided directly by users such as date of birth, gender, working position. As a consequence, it is coming to gain new interest in the task of estimating personal user information directly from the public content produced by them. This task is commonly known as author profiling and aims to make estimations of users descriptive data [20]. Just think of a company that wants to carry out a market survey about its products. It will require information such as age, gender, and other personal and social data about its buyers. Knowing these user details, it can be possible to imagine how targeted advertising campaigns, corrective actions of products, or a better segmentation of the target market can be carried out. Since 2013 [28], the relationship between user characteristics and the language used on social media [20] is widely addressed in PAN campaigns [29] promoted by CLEF (Conference and Labs of the Evaluation Forum). Solutions based on probabilistic representations of text and classic machine learning approaches have proven to be effective in resolving this task. Concurrently, new language model-based linguistic approaches have emerged for many classification and inference tasks [11]. Among them, BERT [6] has proven to be the current state of the art for solving many natural language processing tasks. Given the wide diffusion of these natural language models, the importance of the author profiling task, and the small use of complex deep neural network solutions in this domain, it was decided to propose a deep learning model based on the text representations generated by BERT. We decided to compare the results obtained by our approach with them of SVM, Linear Regression, and Random Forest. Moreover, we compare the computational power required and the execution time of the evaluated approaches to estimate if the differences in predictions justify the high training cost of novel models.

2 Related Work

The author profiling task was widely addressed, each method with a specific focus on some descriptive aspects of the user. Starting from the works in the psychological field [20] that correlated the style of writing with the personality traits of the users, we defined more solutions able to explore more varied connections among written text and user such as related to age, gender, education, and more. Numerous datasets have been released since 2006 [32], starting from data about Emails [8] and Blog articles [31], up to data extracted from social media such as Twitter [3] and Facebook [33]. In particular, since 2013 [28], the number of datasets and solutions released has significantly increased, thanks to the PAN evaluation campaign promoted by CLEF. Looking at the task of author profiling celebrities proposed for PAN 2019 [5] it is possible to observe that although the amount of data available was very large, the number of solutions based on deep neural networks was extremely low (only three experiments with negative results) compared to the number of solutions based on classic machine learning

approaches (Support Vector Machines, Logistic Regression, Random Forests). This observation is a strong indicator of the great difficulty in using deep learning solutions that are scalable in case of big data such as for this task. The winning solution of the 2019 challenge [27] reports the use of SVM and TF-IDF as the most effective solution to predict the traits of fame and occupation, while using logistic regression to predict the year of birth and gender. This approach is considered in this work as a baseline to compare the efficacy of BERT-DNN approach. Moreover, as proposed by Petrik and Chuda [23], we will also evaluate the performances of a random forest classifier with 200 decision trees on TF-IDF vector of the top 10,000 n-grams with the value from one to three. Considering the difficulties in using deep learning approaches in the task, we decided to investigate the issue by using approaches already proven to be effective for other Natural Language Processing tasks [24]. Specifically, the new task-independent language models pre-trained on large amounts of data have proved to be the right way to get a context-aware and efficient word representation for classification tasks. A Task-Independent Language Model is based on the idea of creating a deep learning architecture, particularly an encoder and a decoder, so that the encoding level can be used in more than one NLP task. In this way, it is possible to obtain a decoding level with weights optimized for the specific task (fine-tuning). Following this basic idea BERT (Bidirectional Encoder Representations from Transformers) [7] was trained on a Transformer network with 12 encoding levels, 768-dimensional states and 12 heads of attention for a total of 110M of parameters trained on BooksCorpus [40] and Wikipedia English for 1M of steps. The learning phase is performed by scanning the span of text in both directions, from left to right and from right to left, as was already done in BiLSTMs. Moreover, BERT uses a "masked language model": during the training, random terms are masked in order to be predicted by the net. These peculiarities allow BERT to be the current state of the art language model. We are going to use BERT in its large version to produce contextual word embeddings used as input of our designed deep neural network.

3 BERT-SE: Language Model for Sentence Embeddings

A general-purpose encoder should be able to provide an efficient representation of the terms, their position in the sentence, context, the grammatical structure of the sentence, semantics of the terms. The idea behind language models is that if a model can predict the next word that follows in a sentence, then it is able to generalize the syntactic and semantic rules of the language. In a different way from classic probabilistic word embeddings like word2vec, BERT is developed for the context-aware encoding of terms. Approaches as Word2Vec [17], Glove [21], and FastText [2] suffer from the problem that multiple concepts, associated with the same term, are not represented by different word embedding vectors in the distributional space (the representation is *context-free*). This means that each term has only a single word embedding representation in the distributional space, and different concepts of the same term are not represented. The word embedding created by BERT is dependent on the position of the term in the sentence

(positional embedding), the sentence in which it occurs (sentence embedding), and the co-occurrences of the terms in a window with a bidirectional span (term embedding). Considering this approach, it is therefore clear that in order to embed a single term, it is necessary to encode the entire sentence through the entire BERT encoding pipeline.

Since at the end of each encoding phase, BERT generates a word embedding of the input phrase, it is necessary to decide which one of them to use. In order to solve this problem, several solutions have been proposed in the literature. One of them is to use a strategy similar to the established one for the ELMo language model [22]. Specifically, we will concatenate the results of the last n encoding layers of the model, for example, the last 2. The selection of how many layers to use is a parameter to estimate according to the application domain. Another strategy, proposed in [16] is to use as embedding representative of the whole sentence the value assigned by the model to the $[CLS]$ token, that is the token used as a separator between the sentences adopted in the model for the training phase "next sentence to predict". This solution brings with it the loss of information and does not respect the properties of similarity such that semantically similar sentences have similar representations.

To solve the problems highlighted, we shared the idea of using the approach proposed in [30] based on the use of siamese networks for the generation of semantically meaningful sentence embeddings. The solution proposed by Reimers [30] is based on the use of a modified version of BERT (*BERT-SE*), able to tune the weights of the model so that the sentence embeddings are comparable through a measure of cosine similarity. The model was fine-tuned on the SNLI task (sentences entailment and contradictions) using an objective function able to maximize the similarity among entailed sentences followed by a task of poly-encoders to compute a mean score between all output vectors. In our approach, we used the sentence-transformers library[1] with "bert-large-nli-stsb-mean-tokens" model for the generation of 1024 size sentence embeddings for sentences not longer than 128 tokens. If we incur in phrases longer than this limit, the sentence will be truncated.

4 BERT-DNN for Author Profiling

The model of author profiling proposed in this study (*BERT-DNN*) is based on the synergy between two deep learning classification approaches, the long-short-term memory networks (LSTM) [10] in their bi-directional variation and the convolutional neural networks [13] (CNN) mediated by a max-pooling approach as already adopted in [24]. Moreover, we decided to include, after the Bi-LSTM, a self-attention layer to allow the system to capture distant relationships among words with different weights depending on their contribution to the classification. Figure 1 shows the complete stack of the proposed model.

The first layer of the model has the purpose of accepting as input a set of 150 BERT-SE sentences embeddings representing the last contents generated

[1] https://github.com/UKPLab/sentence-transformers.

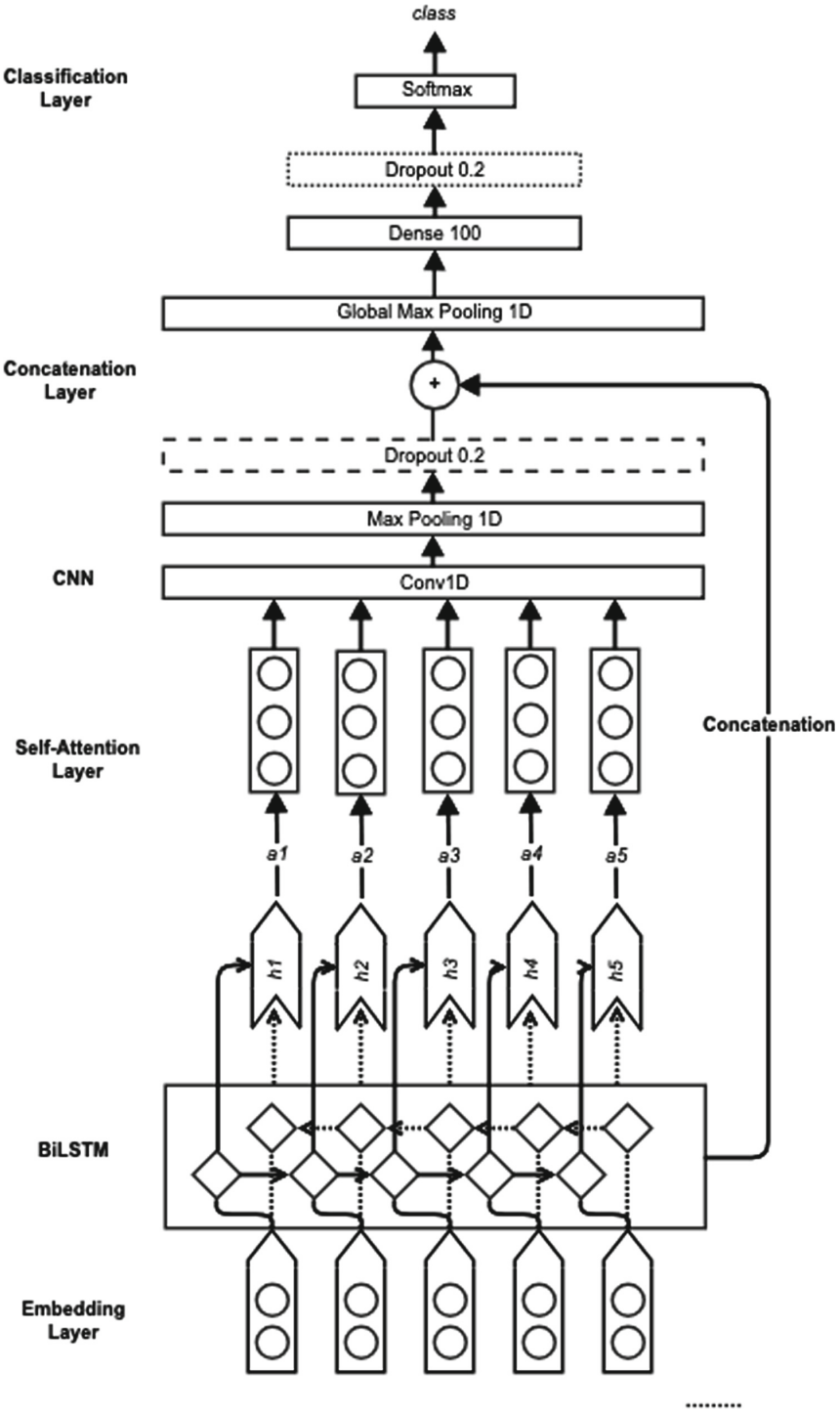

Fig. 1. The architecture of the classification model based on Bi-LSTM, CNN and Self-Attention

by the user. Since some user profiles do not contain enough content, padding with arrays of 1024 zeros will be applied in order to reach the designated input dimension. On the contrary, for the profiles with more content, only the first contents up to the maximum number set will be considered.

Considering the intrinsic sequential relationship between the contents produced sequentially, the contribution made by a recurrent neural network in order to grasp this relationship is evident. LSTM uses the forget gate (hidden neuron) to dynamically scale the weights of its internal "self-loop" depending on the weights learned by the network for previous words provided as input [10]. This step provides the layer a "memory" for considering the relations with the past elements in input. The bi-directional variant considers the relationships among data by both the directions, finally provided as output the concatenation of the links from both the sides. We have configured the LSTM network by setting the value of hidden units to 512 and the dropout value to 0.3. This choice was motivated by the need to reduce the dimensionality of the output of the network so that the operations carried out by the following layers were not computationally too expensive. Moreover, the dropout value was used to reduce, during the learning, the effect of the overfitting on the training data. We have decided to vary also the function of activation used by the net setting it to the hyperbolic tangent function (tanh). This activation function has an S-Shape and produces values among -1 and 1, making layer output more center to the 0. Moreover, it provides a gradient larger than sigmoid function, helping to speed up the convergence [9].

A level of self-attention [4] is added following the LSTM. As well as the attention strategy proposed in [1], self-attention, also known as intra-attention, provides the model ability to weigh the vectors of single words of the sentence differently, according to the similarity of the neighboring tokens. It is possible to say that the level of attention can provide us an idea of what features the network is looking at most during learning and subsequent classification. In particular, we consider an additive self-attention context-aware equal to the whole set of words in input (Eq. 1) [39].

$$\begin{aligned} h_{t,t'} &= tanh(x_t^T W_t + x_{t'}^T W_{t'} + b_t) \\ e_{t,t'} &= \sigma(W_e h_{t,t'} + b_e) \\ a_{t,t'} &= softmax(e_{t,t'}) \\ l_t &= \sum_{t'=1}^{n} a_{t,t'} x_{t'} \end{aligned} \tag{1}$$

where, σ is the element-wise sigmoid function, W_t and W_t' are the weight matrices corresponding to the hidden states h_t and $h_{t'}$; W_e is the weight matrix corresponding to their non-linear combination; b_t and b_e are the bias vectors. The attention-focused hidden state representation l_t of a token at timestamp t is given by the weighted summation of the hidden state representation $h_{t'}$ of all

other tokens at timesteps t. We use the last self-attention implementation for Keras[2].

CNN is a robust neural network ideal for working on data with a shape of grid [12] as a consequence of the convolutional operations performed by the algorithm over adjacent cells. The result of the convolution is a grid more dense and smaller than the previous that captures the hidden relations among cells that fall in the kernel dimension. In our specific case, we applied the CNN layer on the result of the attention algorithm. Such hidden level has a matrix form as a consequence of the vectorial representation supplied by the word embeddings on the tokens in input. In detail, it has the form 150×1024, which allows us to apply a 1D Convolutional network with 1024 filters and 5×5 kernel. We used, as activation function, ReLu that unlike the hyperbolic tangent is faster to calculate [9].

On the top of the CNN layer, we added a Max Pooling function for subsampling the values obtained, reducing the computational load and, the number of parameters of the model. In particular, we used a small 2×2 kernel. On the output of the last max-pooling layer, we applied a dropout function and a dense layer of size for reducing the number of connections inside the model and limiting the effect of overfitting [9]. Dropout is a common regularization technique that, for a defined value of p, sets p fraction of units to 0 at each update during training time. The hidden model obtained until this step has been merged with the output of the previous Bi-LSTM. We apply this operation for letting the model conceptualize both local and long-term features better. After that, we used a max-pooling layer for 'flatten' the results and reduce the model parameter. An analog function of dimensionality reduction is performed by the consequent dense layer and the following dropping function. Finally, another dense layer with a soft-max activation function has been applied for estimating the probability distribution of each class of the dataset.

The model has been trained using the categorical cross-entropy loss function [9], and Adam optimizer for 20 epochs and best models have been used for the classification phase. For the regression task, we substitute the last layer of the model with one using the linear activation function, and the root means squared error as a loss function.

5 Evaluation

The aim of the experimental session is twofold:

- evaluate the efficacy of the here proposed BERT-DNN classification model;
- study the computational cost of BERT-DNN and to compare it with classical machine learning algorithms.

More specifically, in order to achieve the first experimental goal, we performed an experiment where we compare the results obtained by BERT-DNN with them of the two best models presented during the PAN 19 evaluation campaign [5][3].

[2] https://github.com/CyberZHG/keras-self-attention.
[3] https://pan.webis.de/clef19/pan19-web/celebrity-profiling.html.

Moreover, we added another baseline changing the input of our proposed model. In particular, we substitute the BERT-SE embeddings with the TF-IDF vectors of each document considering the most frequent 10000 n-grams of size from one to three, in order to evaluate the contribution of the BERT-SE embeddings to the model. This model is from here referred as TFIDF-DNN.

Our first baseline is the model proposed by Radivchev et al. [27], the winner of the challenge, reports the use of *SVM* and TF-IDF as the most effective solution to predict the traits of fame and occupation, while using *logistic regression* to predict the year of birth and gender. The authors reported the use of a pre-processing pipeline that consists of removing retweets, all symbols except letters numbers, @ and #; replacing URLs, mentions; remove multiple spaces. The user tweets were transformed with a TF-IDF vectorizer, taking into account the top 10,000 features from single words and bigrams. Moreover, they use different *class_weights* depending on the number of labeled examples for each class. During the phase of tuning of hyper-parameters of the classification models, Radivchev et al. report the best performances using the SVM with a *rbf* kernel and $c = 0.1$ for the "fame" aspect and $c = 0.5$ for the "occupation" aspect. Regarding the "gender" aspect, a logistic regression was used with $solver = nnewton_cg$", while a version with standard with $solver =$ "$lbfgs$" was used for the "age" aspect[4].

The second model used as a baseline is the one proposed by Petrik and Chuda [23]. A Random forest with 200 decision trees was chosen as a final classification model. They started with a preliminary pre-processing phase that consists of removing from tweets mentions, letters repeated more than two times, accented letters, and stop-words; URLs have been replaced with a standard <url> tag; emoji have been translated into their word description. Petrik and Chuda us n-grams of size from one to three, as it is commonly and successfully used in a high number of natural language tasks.

The investigation of our second experimental goal has been performed analyzing some simple performance metrics. We measured them during the standard steps performed to create a classification model. First of all, we measure the time in seconds and the need of RAM in MB required by the algorithms to perform pre-processing and sentence encoding. In particular, for these two metrics, we performed a more in-depth analysis varying the amount of data provided as the input of the encoding processes. The same metrics have been also used for the training phase and the prediction of the model on the test set.

Dataset. The evaluation of the model has been performed on the dataset released by Wiegmann et al. [37] for the author profiling competition at PAN 19 [5]. The distinctive characteristic of this dataset is the presence of a new user feature, *the fame of social networks*, that has been added at the list of the other more common user descriptive features. Moreover, this dataset has been robustly validated through Wikidata. The user profiles have been linked with it, thank the peculiarity that they were referring to the profiles of famous people

[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

(celebrities). The dataset contains 33,836 celebrities with up to 3,200 tweets each and 156,411,899 tweets in total (around 3 billion words). The task organizers do not publicly provide the dataset used for the testing phase during the challenge. Consequently, we started with the idea to split the whole dataset into portions 70% training, 10% validation set, 20% test set. Unfortunately, starting the process of sentence conversion into word-embeddings, using the BERT-SE strategy described in Sect. 3, we noted that the processing took a too long time (around 50000 tweets/hour). As a consequence, we randomly selected 6000 celebrities for the training phase, 1500 for the validation and 2250 for the test, limiting the number of tweets for each account at 150, for a total number of tweets equal to 1462500.
Into the dataset, the following values are possible for each of the user traits to predict:

```
fame       := {rising, star, superstar}
occupation := {sports, performer, creator,
               politics, manager, science,
               professional, religious}
birthyear  := {1940, ..., 2012}
gender     := {male, female, nonbinary}
```

The prediction performance for $T \in \{gender, fame, occupation\}$ is measured using the macro-averaged multi-class F1-score. The *"birtyear"* is considered as correct if it is within an m-window of the true year, where m increases linearly from 2 to 9 years with the true age of the celebrity in question: $m = (-0.1 * truth + 202.8)$ as described in the official competition [5].

The code[5] has been run on a Google Colab[6] Python 3 environment equipped with 25 GB of RAM, a GPU (a single 12 GB NVIDIA Tesla K80 GPU) and unlimited disk space on Google Storage Bucket Platform.

Discussion of Results. The results in Table 1 shows how the approach BERT based (BERT-DNN) is the best compared with the three baselines. As described in Sect. 2, the approaches we are comparing are the two best results during the PAN 2019 celebrity profiling competition and a variation of the here proposed BERT-DNN method that uses documents TF-IDF as input. For all the four classification tasks the F1 score obtained by BERT-DNN is around 2% better than the one of competitors. This result highlights the optimal performance of the neural model compared with the traditional machine learning approach. The differences in scores have been statistically validated for each pair of models using Wilcoxon Signed-Ranked Test. We obtain that the differences among BERT-DNN and all the other approaches are statistically significant for $p < 0.05$. The best performances of BERT-DNN are not surprising. It is well known in the literature that the new language models such as BERT, RoBERTa [14], ERNIE [34], are actually the best resources to use for formalizing the relations among

[5] The source code of the project can be found at the following GitHub repository: https://github.com/marcopoli/ICCSA2020_author_profiling.
[6] https://colab.research.google.com/.

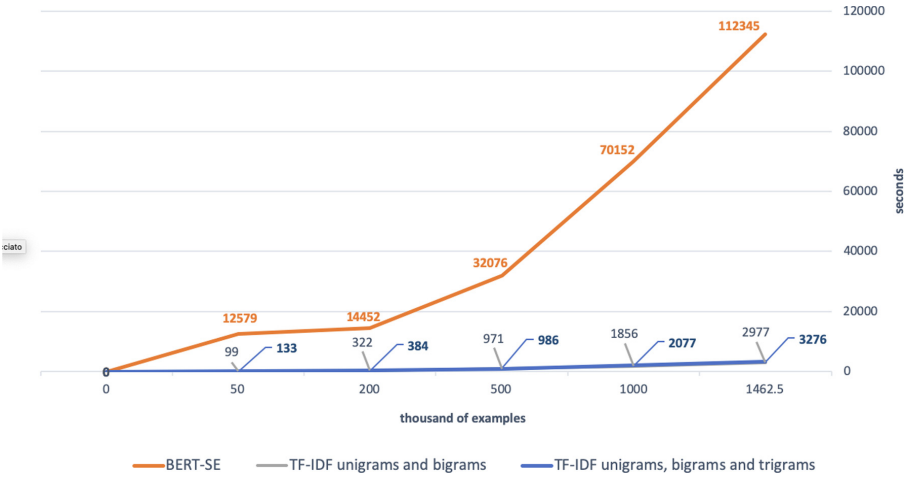

Fig. 2. In the figure it is reported the variation of the encoding time varying the number of examples provided to the function.

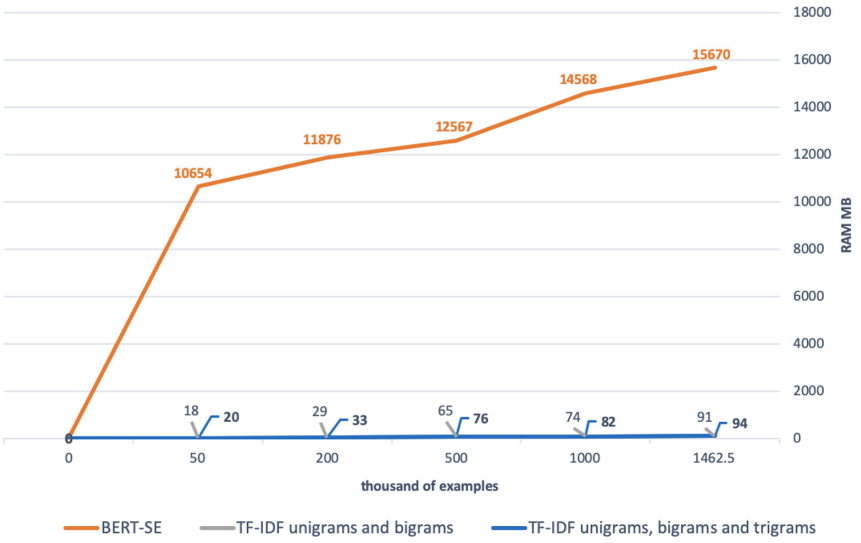

Fig. 3. In the figure it is reported the variation of the RAM need for encoding phrases varying the number of examples provided to the function.

words and their semantical meaning. Analogs results have been obtained in many other tasks of natural language understanding [36]. In particular, in our experiment, the relevance of BERT based embeddings has been demonstrated by the differences obtained in our model using the TF-IDF strategy as the input of the DNN. Indeed, it is possible to observe that these results obtained by the TFIDF-DNN are always lower of them of BERT-DNN. This encouraging results

Table 1. F1 scores obtained for the prediction of each of the author's descriptive features. Results market with a $*$ are statistically significant for p <0.05 using a Wilcoxon Signed-Ranked Test.

Task	BERT-DNN	TFIDF-DNN	SVM (rbf) + LinearRegression [27]	Random Forest trees = 200 [23]
Gender F1-score	**0.89415***	0.87523	0.86738	0.65980
Age F1-score	**0.83221***	0.813452	0.80489	0.69784
Fame F1-Score	**0.77840***	0.75712	0.74420	0.67321
Occupation F1-Score	**0.72733***	0.71095	0.70871	0.65978

Table 2. In the table are reported some statistics about the computational cost of the three approaches here discussed.

Metric	BERT-DNN	SVM (rbf) + LinearRegression [27]	Random Forest trees = 200 [23]
Text encoding time (s)	BERT-SE 112345	TF-IDF n-grams; n = 1, 2 **2977**	TF-IDF n-grams n = 1, 2, 3 3276
Text encoding space (MB)	BERT-SE 15670	TF-IDF n-grams; n = 1, 2 **91**	TF-IDF n-grams n = 1, 2, 3 94.7
Learning Time (s)	TPU-V2 934	CPU 1084	CPU **669**
Learning Space (MB)	RAM 18346	RAM **789**	RAM 922
Prediction Time (s)	TPU-V2 **12**	CPU 185	CPU **125**
Prediction Space (MB)	RAM 12561	RAM **649**	RAM 856

support our hypothesis about the importance of using a BERT based approach for sentence embedding, such as BERT-SE, for obtaining more accurate results when we deal with a text classification task, e.g., author profiling.

On the other hand, observing the analysis of complexity described in Table 2, it is easy to note how the computational time and space of the BERT-DNN model are very high in total compared with those of SVM and Random Forest. Classical models can be run on a standard computer with few RAM (8GB are more than enough) and a standard CPU. A BERT-based approach requires, instead, a TPU (patented by Google) and a much high performer machine. Moreover, using a classic SVM approach, it could be possible to train the model on a much larger dataset without encounter the problems observed using BERT-DNN.

Moreover, observing the Fig. 2 it is possible to note how the BERT-SE encoding takes a quite flat time of execution until it reaches a threshold of about 200,000 examples. After that, the time of encoding is increasing linearly. This behavior is due to the time necessary at the startup to load into the computer memory the BERT-SE pre-trained model. After that, the model starts to work on the sentences encoding normally, showing an increase of time of execution proportionally to the number of sentences to process. A slight reduction of performances is observable in the same figure when the model overcomes the threshold of 1,000,000 examples. In this scenario, the large amount of memory required for storing temporal data negatively affects also the execution time. The behavior of the TF-IDF vectorizer is quite constant without significant differences when working with only unigrams and bigrams or also with trigrams. The behavior of BERT-SE is easier to observe in Fig. 3 where it is shown the variation in consumption of RAM during the encoding phase. In an initial step, BERT-SE consumes a large amount of RAM for loading into the computer memory its pre-trained model. After that, the amount of ram required is increasing linearly. Also in this case, the behaviour of the TF-IDF vectorizer is constant requiring a low amount of RAM without significant distinction if it is working with unigrams, bigrams or trigrams.

Taking into consideration both the aspects, the model performances in complexity and accuracy, we can consider the increment in performances of the BERT-based approach too much low to be justified by the enormous quantity of computational power need for running it. Our hypothesis about the increase in performances using a BERT based approach is confirmed, but the large amount of computational power need for running it is still not convincing. This can cause significant limits in research about the topic forcing those who do not have the appropriate resources to settle for reduced datasets or lower precision of the model. In this regard, we would like to suggest to the community a further research effort to make these models more straightforward and more affordable for everyone in a future perspective of using deep learning in everyday activities of common use.

6 Conclusion

The prediction of descriptive features of a digital profile is a task of author profiling that is continuously gathering increasing attention. In this work, we proposed an approach based on the latest natural language model, i.e., BERT-SE and an LSTM-CNN deep neural network. We compared the accuracy of this model with classic approaches of machine learning that won the PAN 19 competition about author profiling. The results are encouraging if we only consider the F1 score obtained on the celebrity profiling dataset, but are very disappointing if we also consider the computational power need for running the BERT-DNN model. Novel strategies that consume less computational power should be investigated more in the future, and when less complex strategies could obtain similar performances in the scenario of application, these should be adopted. In this regard,

we would like to encourage the further analysis of trade-off between accuracy and performances because a small improvement could, very often, be not justified by the high increase of computational power need for running the model. Moreover, we would like to support the idea that BERT is not always the best solution for any application scenario, as often claimed. In particular, a trade-off analysis should always be reported into new researches when a new state of the art result is claimed.

Acknowledgment. This research has been funded by Regione Puglia under the programme "INNOLABS" Sostegno alla creazione di soluzioni innovative finalizzate a specifici problemi di rilevanza sociale. POR Puglia FESR – FSE 2014-2020. Asse prioritario 1 – Ricerca, sviluppo tecnologico, innovazione. Azione 1.4.b - Project: Feel@Home cod. UIKTJF3.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
3. Burger, J.D., Henderson, J., Kim, G., Zarrella, G.: Discriminating gender on twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1301–1309. Association for Computational Linguistics (2011)
4. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733 (2016)
5. Daelemans, W., et al.: Overview of PAN 2019: author profiling, celebrity profiling, cross-domain authorship attribution and style change detection. In: Crestani, F., et al. (eds.) 10th International Conference of the CLEF Association (CLEF 2019). Springer, September 2019. http://ceur-ws.org/Vol-2380/
6. 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)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. https://www.aclweb.org/anthology/N19-1423
8. Estival, D., Gaustad, T., Pham, S.B., Radford, W., Hutchinson, B.: Author profiling for English emails. In: Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics, pp. 263–272 (2007)
9. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep Learning, vol. 1. MIT Press, Cambridge (2016)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Jing, K., Xu, J., He, B.: A survey on neural network language models. arXiv preprint arXiv:1906.03591 (2019)
12. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)

13. LeCun, Y., et al.: Generalization and network design strategies. In: Connectionism in Perspective, pp. 143–155 (1989)
14. Liu, Y., et al.: Roberta: a robustly optimized Bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
15. López-Monroy, A.P., Montes-y Gómez, M., Escalante, H.J., Villasenor-Pineda, L., Stamatatos, E.: Discriminative subprofile-specific representations for author profiling in social media. Knowl. Based Syst. **89**, 134–147 (2015)
16. MacAvaney, S., Yates, A., Cohan, A., Goharian, N.: Cedr: contextualized embeddings for document ranking. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1101–1104 (2019)
17. 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)
18. Musto, C., Semeraro, G., Lovascio, C., de Gemmis, M., Lops, P.: Myrror: a platform for quantified self and holistic user modeling. In: Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, pp. 215–216 (2018)
19. Pang, B., Lee, L., et al.: Opinion mining and sentiment analysis. Found. Trends® Inf. Retrieval **2**(1–2), 1–135 (2008)
20. Pennebaker, J.W., Mehl, M.R., Niederhoffer, K.G.: Psychological aspects of natural language use: our words, our selves. Annu. Rev. Psychol. **54**(1), 547–577 (2003)
21. 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)
22. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
23. Petrik, J., Chuda, D.: Twitter feeds profiling with TF-IDF notebook for PAN at CLEF 2019, vol. 2380 (2019)
24. Polignano, M., Basile, P., de Gemmis, M., Semeraro, G.: A comparison of word-embeddings in emotion detection from text using BiLSTM, CNN and self-attention. In: Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization, pp. 63–68 (2019)
25. Polignano, M., Basile, P., Rossiello, G., de Gemmis, M., Semeraro, G.: Learning inclination to empathy from social media footprints. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, pp. 383–384 (2017)
26. Polignano, M., de Gemmis, M., Narducci, F., Semeraro, G.: Do you feel blue? Detection of negative feeling from social media. In: Esposito, F., Basili, R., Ferilli, S., Lisi, F. (eds.) Conference of the Italian Association for Artificial Intelligence, pp. 321–333. Springer (2017)
27. Radivchev, V., Nikolov, A., Lambova, A.: Celebrity profiling using TF-IDF, logistic regression, and SVM notebook for pan at CLEF 2019, vol. 2380 (2019)
28. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: Overview of the author profiling task at pan 2013. In: CLEF Conference on Multilingual and Multimodal Information Access Evaluation. pp. 352–365. CELCT (2013)
29. 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. In: Working Notes Papers of the CLEF, pp. 1613–1673 (2017)
30. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)

31. Rosenthal, S., McKeown, K.: Age prediction in blogs: a study of style, content, and online behavior in pre-and post-social media generations. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 763–772. Association for Computational Linguistics (2011)
32. Schler, J., Koppel, M., Argamon, S., Pennebaker, J.W.: Effects of age and gender on blogging. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, vol. 6, pp. 199–205 (2006)
33. Schwartz, H.A., et al.: Personality, gender, and age in the language of social media: the open-vocabulary approach. PLoS ONE **8**(9), e73791 (2013)
34. Sun, Y., et al.: Ernie 2.0: a continual pre-training framework for language understanding. arXiv preprint arXiv:1907.12412 (2019)
35. Wachter, S.: Normative challenges of identification in the internet of things: Privacy, profiling, discrimination, and the GDPR. Comput. Law Secur. Rev. **34**(3), 436–449 (2018)
36. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: a multitask benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461 (2018)
37. Wiegmann, M., Stein, B., Potthast, M.: Celebrity profiling. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2611–2618 (2019)
38. Zhang, Y., Pennacchiotti, M.: Predicting purchase behaviors from social media. In: Proceedings of the 22nd international conference on World Wide Web, pp. 1521–1532 (2013)
39. Zheng, G., Mukherjee, S., Dong, X.L., Li, F.: OpenTag: open attribute value extraction from product profiles. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1049–1058. ACM (2018)
40. Zhu, Y., et al.: Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 19–27 (2015)