UOBIT @ TAG-it: Exploring a Multi-faceted Representation for Profiling Age, Topic and Gender in Italian Texts.

Roberto Labadie Tamayo, Daniel Castro Castro and Reynier Ortega Bueno
Computer Science Departament, University of Oriente
Santiago de Cuba, Cuba
roberto.labadie@estudiantes.uo.edu.cu,
{danielcc, reynier}@uo.edu.cu

Abstract

English. This paper describes our system for participating in the TAG-it Author Profiling task at EVALITA 2020. The task aims to predict age and gender of blogs users from their posts, as the topic they wrote about. Our proposal combines learned representations by RNN at word and sentence levels, Transformer Neural Nets and hand-crafted stylistic features. All these representations are mixed and fed into a fully connected layer from a feed-forward neural network in order to make predictions for addressed subtasks. Experimental results show that our model achieves encouraging performance.

The growing integration of social media with people's daily live has made this medium a common environment for the deployment of technologies that allow the retrieval of useful information in the development of business activities, social outreach processes, forensic tasks, etc. That is because people frequently upload and share content in these media with various purposes such as socialization of points of view about some topic or promotion of personal business, etc. The analysis of textual information from such data, is one of the main reasons why researches become trending on the Natural Language Processing (NLP) field.

However, the fact that this information varies greatly in terms of its format, even when it comes from the same person, besides textual sequences are unstructured information, make challenging the process of analyzing it automatically. Author Profiling (AP) task aims at discovering different marks or patterns (linguistic or not) from texts, that allow a user to be characterized in terms of their age, gender, personality or any other demographic attribute.

Many forums, due to the applicability of AP, share tasks directed to mining features that in general way, predict that valuable information. Those tasks commonly make special focus on popular languages such as English and Spanish. Nevertheless, other languages are explored on important forums too, that is the case of EVALITA [1], this one, promoting analysis of NLP tasks in the Italian language. Among the challenges from its last campaign *EVALITA 2018* was the AP (in terms of gender) task *GxG* (Dell'Orletta and Nissim, 2018), exploring the gender-predicting issue.

The analysis of age, gender and the topic a text is related with, are tasks well explored and the most approaches employ data representation based on stylistic features, n-gram representations and/or words embedding combined with Machine Learning (ML) methods like Support Vector Machine (SVM) and Random Forest (Pizarro, 2019). Also some authors by using Deep Learning (DL) models like Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM) combined with stylistic features (Aragón and López-Monroy, 2018) (Bayot and Gonçalves, 2018) have yield encouraging performances.

In this work we address precisely, the automatic detection of gender and age of the authors, besides the identification of the prevailing topic on textual information from blogs. Also, we describe our developed model for participating on *TAG-it: Topic, Age and Gender prediction for Italian*[2] (Cimino A., 2020) task at *EVALITA 2020* (Basile et al., 2020).

Having in account the proved ability of DL

[1] http://www.evalita.it/

[2] https://sites.google.com/view/tag-it-2020

models to learn abstract depictions that are omitted in hand-crafted features engine methods, our approach is mainly based on them, particularly on Bi-LSTM and Transformer Nets (Vaswani et al., 2017). We combine the feature representations learned by DL models, with hand-crafted ones based on Term Frequency-Inverse Document Frequency *(tf-idf)* and stylistic features.

This paper is organized as follow: in the next section a brief description about the different subtasks of *TAG-it* task. Next, we present our proposal. Specifically, we describe the data preprocessing as well as the DL methods and features used for depicting this data. Finally, the experimental setting, the experiments conducted and the results achieved.

1 TAG-it Tasks

Three sub-task have been proposed on TAG-it task.

- **subtask 1**: Toward to predict the gender, the age (as an age range, eg: 20-29) and the topic mentioned by the author given a collection of texts written by him/her from a blog, all this three dimensions at once.
- **subtask 2a**: For predicting gender.
- **subtask 2b**: For predicting age.

For these tasks a training corpus of texts written by blogs users, with possibly multiple posts per user, was provided. Each user information (i.e posts per user) varies in terms of its length and quantity, and the data for each subtask is unbalanced mainly for gender and topic prediction tasks, which place some complexity degree for the training stage of the models for these classification tasks.

2 Our Proposal

Deep Learning methods are capable to learn and project relationships between elements within textual information which are beyond the human abstract comprehension. Therefore the use of just hand-crafted representations may omit some important patterns on textual information analysis.
However, stylistic and linguistic features have proved to be good marks to determine some author characteristics. Within the used DL models on AP field, are the LSTM (Labadie-Tamayo et al., 2020) and the Transformers Neural Nets, which rely on two different paradigms. The first ones analyses the information sequentially, token by token whereas the second ones analyze all these tokens at once, relating every one with respect to each other. The opposite behavior of these two architectures implies learning different patterns which individually have proved to be an accurate way to synthesize the information.
We hypothesize that making an ensemble of these deep representations and fusing it with hand-crafted ones as we show on Figure. 1 could yield encouraging results on the proposed tasks.

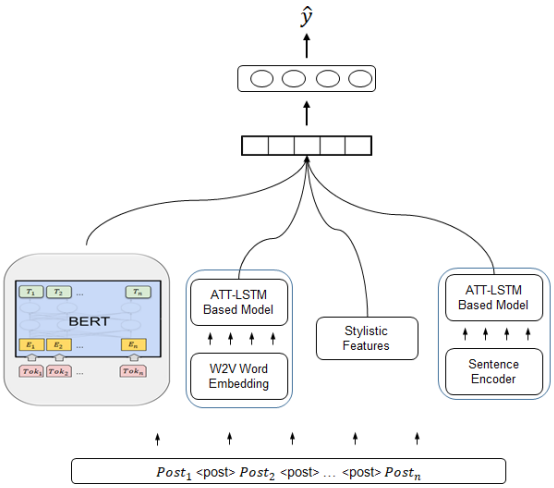

Figure 1: Representations Ensemble

The first representation *(Transformer Block)* based on Bidirectional Representation from Transformers (BERT) Architecture (Devlin et al., 2018). The second based on LSTM (Hochreiter and Schmidhuber, 1997) neural nets with self attention mechanism (Att-LSTM) by using words embedding *(Recurrent Word-Level Block)*. The third one, a condensed representation based on the combination of stylistic features and a vector with the *tf-idf* computation of some keys tokens from the text *(Stylistic Block)*. Finally *(Recurrent Sentence-Level Block)*, another representation based on Att-LSTM, but at this time, analyzing the sequence information at sentence level.
All these representations are concatenated and fed into a dense layer, by using Leaky Rectified Linear Unit (Leaky ReLU) activation function, to synthesize the extracted information on each block and its output vector goes to a softmax dense layer which have the same number of neurons as classes on the analyzed task, in order to make the predictions.

For dealing with the three classification tasks we used the same architecture, but trained separately for each of them, with different targets attending to the task.

2.1 Preprocessing

In the preprocessing stage we concatenate the posts corresponding to the same user, in order to treat them as only one super-document, but between each post we place a tag i.e ⟨ post ⟩ denoting the ending-beginning of them. Afterwards, the numbers and dates are recognized and replaced by a corresponding wildcard which encodes the meaning of these special tokens. Then, the text is tokenized and morphologically analyzed by means of FreeLing (Padró and Stanilovsky, 2012).
For computing the stylistic and *tf-idf* vectors as for feeding the deep models on prevailing topic detection task, we removed the stop words from the document and lemmatized the tokens to their canonical form.

2.2 Transformer Block. BERT

BERT (Bidirectional Encoder Representations from Transformers) is an architecture resulting of applying a bidirectional training to the attention model Transformer, designed for language modeling. The Transformer model has two mechanisms, the first one, known as the encoder, which is fed with the text and finds out an encoded representation for the sequence. The second one, the decoder, produces the predicted tokens for language modeling one at time, having in account the encoder's output and the previous predicted tokens on each time step.
The main advantage of this transformer models w.r.t. traditional sequential architectures like Gated Recurrent Unit (GRU) (Cho et al., 2014) is that instead of analyzing the textual information in one or another direction (e.g. right to left or left to right) it takes in account the entire information at once by using an attention mechanism, which relates each word on the text with its surrounding context.
Since the goal of BERT is to generate a language representation, only the encoder mechanism is necessary. It is structured with transformer blocks connected sequentially and each transformer block is composed by attention heads working in parallel. These transformer blocks give to their subsequent layer one representation for each element of the input text, but these representations correlates the entire input context.
The original BERT model is trained with two subtasks, one of them consisting on predict some masked words from a sentence and the other one consisting on predict if two sentences are consecutive in the given corpus text.
For the *TAG-it* tasks we employed a pre-trained BERT model on a multilingual corpus (multilingual_L-12_H-768_A-12)[3] (Turc et al., 2019), which is fed with the super-document sequence. From this model we just used the first two transformer blocks and as its output we keep the first and last vectors from the input sequence encoding, which are concatenated.
Also we applied fine tuning on BERT, adding an intermediate dense layer of 64 units by using Leaky ReLU activation function, and taking as target for training a multitask focus trying to make predictions for age, topic and gender tasks at once.

2.3 Recurrent Word-Level Block

The second representation block of our system is based on LSTM nets. This block takes as input a sequence of the preprocessed text information, which is fed into an embedding layer, set up with fixed weights from FastText (Grave et al., 2018) pretrained word embedding[4], obtaining from each word of the sequence a vectorial representation.
The textual sequence is provided with relevant or not information with respect to the task in analysis. In order to highlight the most important elements for encoding the message instead of making the network pays attention to all elements alike, the embedding layer output tokens are scored by its relative importance over the other elements on its context with Scaled Dot-Product Attention Mechanism (Vaswani et al., 2017). Then, the new scored sequence is fed into a Bidirectional-LSTM (BI-LSTM) (Schuster and Paliwal, 1997) layer with 64 neurons which perform two analysis over this sequence, in forward and backward directions, for detecting not just relations of an element with the previous ones, but also with the elements that appear after it. Afterwards, the hidden states from the Bi-LSTM layer are considered as a new sequence, which is fed into another LSTM with 64 neurons too, taking from its output just the last hidden state, which represents the Recur-

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

[4] https://fasttext.cc/docs/en/crawl-vectors.html

rent Word-Level Block encoding.
For training this block we applied dropout (Srivastava et al., 2014) to the neurons of the attention and LSTM layers in order to improve the generalizing capability of the model.

2.3.1 Scaled Dot-Product Attention

This attention function at first, maps for each sequence token three representations (the query and a key-value pair) for computing a compatibility index between every pair of elements. Afterwards, for each token t_i is evaluated its compatibility w.r.t every other sequence token t_j by relating its query vector q_i with all the keys k_j, then these compatibilities c_{ij} are normalized with a softmax function and used for scoring the value vectors v_j in front of that specific query. Finally, the attention based representation for t_i is computed as the weighted sum of these pondered values vectors. This computation is defined as follows:

$$Attention(Q, V, K) = softmax(\frac{Q \times K^T}{\sqrt{d_k}}) \times V \tag{1}$$

Where $Q, K \in \Re^{n \times d_k}$ and $V \in \Re^{n \times d_v}$ are matrices, which, on every row contain for query, key and value respectively the mappings of the sequence tokens, n corresponds to the length of the sequence and d_k, d_v to the dimension of mapping vectors for key and value respectively.

2.3.2 LSTM

LSTM networks are a special kind of RNNs, which are specialized on analyzing sequential data. These have a main cell unit (the recurrent unit) which explores the data sequence one element at each time step (left to right order). This network shares the information captured in previous steps, for computing the new hidden state at the current time step. Inside the main cell is contained a gate structure that informs to the network which information preserve or forget from the hidden sates of previous time steps for the current computation.

2.4 Stylistic Block. Stylistic Features

The Representation based on stylistic features is twofold; in one side we consider for characterizing a user attending to some classification task, a vector containing the *tf-idf* of a set of key tokens from the text and on the other side we construct a statistical style features vector which captures information from distinct lexical and syntactical linguistic layers.
For constructing the first one we used a feature selection approach which score every term employed by users corresponding to some category within a classification task and then are selected the more relevant ones.
For scoring the tokens we use IG (Sebastiani, 2002) standing for Information Gain, which takes into account the presence of a term in a category as well as its absence. The information gain of a term t in a class C is defined as:

$$IG(t, C) = \sum_{c \in \{C, \bar{C}\}} \sum_{x \in \{t, \bar{t}\}} P(x, c) \log_2 \frac{P(x, c)}{P(x)P(c)} \tag{2}$$

In this formula, probabilities are interpreted on an event space of documents (e.g. $P(\bar{t}, C)$ indicates the probability that, for a random document d, term t does not occur in d and d belongs to category C).
Once computed the IG for every term which belongs to documents of the class c_i, the $\frac{500}{l_c}$ tokens with highest IG are chosen for characterizing this class , where l_c is the number of the task classes. Finally a $500 - dimensional$ vector is constructed where its components are computed as the *tf-idf* of the representative terms from every class.

The second representation is computed independently of the addressed task as a $12 - dimensional$ vector where its components are real numbers corresponding to statistical values from lexical and syntactical linguistic layers (e.g sentence, paragraph, syntactic layers) such as:

- Paragraph layer: Standard deviation of the sentences' length written by the user.
- Text layer: Number of stop words used.
- Sentence layer: Average of words' length.
- Syntactic layer: Proportion of nouns over adjective.

These two representations are combined and fed into a 64-neurons dense layer to synthesize the information and later being fused it with the other blocks representations.

2.5 Recurrent Sentence-Level Block

This block shares the same structure with the *Recurrent Word-Level Block*, but instead to be

fed with a sequence composed by word representations provided by a word embedding layer, it is fed with a sequence resulting of encoding each super-document's sentence by means of an encoder with a similar structure as the first analyzed *Transformer-Block* .

For this Recurrent Sentence-Level Block, we trained the sentence encoder with the same multi-task focus as in the *Transformer-Block* , but aiming to predict for each sentence from a document the annotated characteristics (i.e age and gender) of the user who it belongs to and the topic of its surrounding text. Then we encode all the sentences from the super-document composed by the user's posts, and we considered them as tokens from a sequence at sentence level. Afterwards, that sequence is fed into a model with the same structure Att-Bi-LSTM as the *Recurrent Word-Level Block* taking from this, as the user's profile encoding, the last hidden state from the second LSTM layer as in the Word-Level block.

3 Experiments and Results

The dataset used in this work was the one provided by the task organizers. This dataset is unbalanced, mainly for gender classification task, where the male class represents the 82.6% of the examples.
In order to prevent a biased training of the model we applied a class-weighting method, scoring the computed loss for every examples having in account the class which it belongs to (i.e for examples from male class we give to the computed loss a weight of 0.3 whereas for female examples we pondered the loss to 0.7) this makes that when parameters are updated by means of the gradients, the models pays more attention to the most weighted class, specifically to the under-represented class.
We pretrain the Transformer models from the *Transformer Block* and the sentence encoder of the *Recurrent Sentence-Level Block* independently of the entire model and then we fixed the learned weights.
For fine tuning these BERT models we employ Adam Optimizer, using categorical cross-entropy loss function for every output layer, since we applied multi-task learning over two epochs. The learning rate for this training was set up to a low value (lr=1e-5) since we wanted to keep the parameters learned from the original train with an enormous data as more as possible, while we made the model focus on our addressed tasks, also we set the decay = 2e-3 to the learning rate scheduler.

We evaluate and select the hyper-parameters as the representation and features that we used for our model by using a cross-validation method to obtain a more realistic an unbiased performance evaluation, making 5 splits for validation. On each cross validation step, the dataset was split in 20% for validation and 80% for training, keeping the distribution of examples relative to the split size.
The performance of the model on training stage was evaluated independently for each subtask by using different combinations of representations from Recurrent Word-Level Block (RNN-W), Recurrent Sentence-Level Block (RNN-S), Transformer Block (T) and Stylistic Block (STY).
For age and gender prediction we employed Micro-F1 metric whereas for topic prediction we used accuracy metric for the evaluation. In Table. 1 we summarize the results obtained in terms of the average of these metrics in cross-validation training.
As we can see, assembling the three deep representations with the stylistic one, yield a good performance in all cases through the cross-validation process. However, the stylistic representation had a soft negative influence on gender prediction task.

Table 1: Model Performance on training data.

Model	**Age** AVG-F1	**Gender** AVG-F1	**Topic** Acc
RNN(S+W)-STY-T	**0.378**	0.941	**0.935**
RNN(S+W)-T	0.203	**0.946**	0.885
RNNS-STY-T	0.348	0.940	0.931
RNNW-STY-T	0.339	0.919	0.903

Regarding the official results, we submitted 3 runs as **UOBIT** team, on each of them we employed the representations learned by the Transformer and Stylistic Blocks by tuning the use of the Recurrent Blocks' encode, as shown on Table. 2.

After the evaluation phase we try to remove the stylistic features based representation and we found out that this representation, possibly be-

Table 2: Model Performance on test data.

run	Model	Subtask 1		Subtask 2a	Subtask 2b
		Metric 1	Metric 2	Micro-F1	Micro-F1
run-1	RNN-W T STY	0.686	0.251	0.852	0.278
run-2	RNN-S T STY	0.674	0.243	0.883	0.370
run-3	RNN-W RNN-S T STY	**0.699**	**0.251**	0.893	0.308
		Unofficial			
-	RNN-W RNN-S T	0.680	0.248	**0.898**	**0.4680**
-	RNN-W RNN-S	0.667	0.243	0.893	0.369
-	T	0.436	0.067	0.835	0.283

cause of it introduces some noise, makes the model to have a worst performance, at least on those tasks related to the author attributes (i.e gender and age) corresponding to task 2a and task 2b. We think that noise introduced by these features mainly comes from the fact that they are computed based on key tokens from the text, these tokens may suggest to the model that texts with same topic belongs to the same class within gender or age classification task.

The performance of our system just by using the deep representations of the Recurrent and Transformer Blocks, yield a performance of 0.4606 under F1 metric on subtask 2b which improves the ones reached by the best team of 0.409, whereas this same combination improves our best official run on subtask 2a. These results are shown on Table. 2 under the row named Unofficial.

4 Conclusions

In this paper we described our system for participating in the TAG-it Author Profiling task at EVALITA 2020. Our proposal is based on an ensemble of RNN, Transformer Neural Nets and hand-crafted stylistic features. The system receives as input a user's profile textual information as an only one super document (sequence), this information is encoded in four different ways, the first one by a Transformer Block, specifically a fine tuned and reduced BERT model, the second one, by a Recurrent Block based on an Attention-Bi-LSTM model analyzing the information at word level, the third one by a feature representation based on the combination of *tf-idf* information and stylistic features extracted from the text. Finally the fourth one by the same recurrent structure as in the Recurrent Worf-Level Block, but analyzing the information at sentence level.

This four representations are mixed and fed into a dense layer for synthesize them and its output is received by another dense layer which classify this profile taking into account the classes from the addressed subtask.

The results shown that considering both the stylistic representation and the deep representations learned by Recurrent and Transformer models we obtain the best effectiveness based on the accuracy measure for the task related to the topic classification, but this behavior changed for age and gender classification, due to the relationship of syntactic structures of the text with the topic that the user's posts are related to. We think that excluding the stylistic features or at least those related to the frequency of tokens from the text, could be a way to increase the effectiveness of the ensemble, mainly on the age detection subtask. Also analyzing the content of the posts at character level, due to the informal text origin, would solve the problem of missidentification of some key words within te text. We would like to explore these ideas in future work.

References

Mario Ezra Aragón and A-Pastor López-Monroy. 2018. A straightforward multimodal approach for author profiling. In *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)*.

Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro. 2020. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

Roy Khristopher Bayot and Teresa Gonçalves. 2018. Multilingual author profiling using lstms: Notebook for pan at clef 2018. In *CLEF (Working Notes)*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Nissim M. Cimino A., Dell'Orletta F. 2020. Tag-it@evalita2020: Overview of the topic, age, and gender prediction task for italian.

Felice Dell'Orletta and Malvina Nissim. 2018. Overview of the evalita 2018 cross-genre gender prediction (gxg) task. *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12:35.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Roberto Labadie-Tamayo, Daniel Castro-Castro, and Reynier Ortega-Bueno. 2020. Fusing Stylistic Features with Deep-learning Methods for Profiling Fake News Spreader—Notebook for PAN at CLEF 2020. In Linda Cappellato, Carsten Eickhoff, Nicola Ferro, and Aurélie Névéol, editors, *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org, September.

Lluís Padró and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May. ELRA.

Juan Pizarro. 2019. Using n-grams to detect bots on twitter. In *CLEF (Working Notes)*.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.