Gender, Age, and Dialect Recognition using Tweets in a Deep Learning Framework -Notebook for FIRE 2019

Chanchal Suman¹, Purushottam Kumar², Sriparna Saha¹, and Pushpak Bhattacharyya¹

 ¹ Dept. of Computer Science & Engineering, Indian Institute of Technology Patna, Patna, India
² Dept. of Computer Science & Engineering, National Institute of Technology

Durgapur, Durgapur, India

{1821cs11, sriparna, pb}@iitp.ac.in , pk.17u10625@btech.nitdgp.ac.in

Abstract. Social media sites are a rich platform for a user-generated text that can be used to identify different aspects of the authors. Age, gender, dialect, region, are different aspects of an author, which can be identified with the proper mining of these contents. This profiling provides a way of identifying anonymous users. Recognizing the profile of an anonymous user help in indirect recognition of the identity of the user. In this notebook, we describe the working of our author profiling software submitted for FIRE 2019 which recognizes the gender, age, and dialects of Twitter users in the Arabic language. We have used Long short term memory neural network and some hand-crafted features for recognizing the age, gender, and dialects of the author of a tweet³.

Keywords: Dialects, LSTM, Aravec, emoji, emoticons

1 Introduction

Author profiling is the technique to identify the different aspects of the author from a given text. It differentiates among the different classes of an author by studying their writing style and the words used in their text. It also shows how the behavior viewpoint is used to recognize the different classes of an author. It tells about the uses of different writing skills and how the language is shared by a different author in their text. The textual information based on different features and styles helps identify the author's profile based on different aspects such as gender, age, and dialect[7]. The focus of this task is to identify the gender, age and the dialect variety of Arabic Twitter users. these information about an anonymous text can be used in detecting criminal in cyber-forenciscs. During investigation, it is very tough to get the idea about the real guilty. This type

³ Copyright ©2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). FIRE 2019, 12-15 December 2019, Kolkata, India.

2 Chanchal et al.

of analysis helps the authority to conclude about the traits of the guilty and helps in chasing the possible suspects. Gender is an important aspect of a user. if it is detected correctly, then it would be very helpful in selecting the possible suspects. Similarly, the age and the language variety too. These applications of such analysis motivated us to do research in this area. There have been several papers by too many researchers over the years studying on the topic of Author Profiling in text. The text used in these papers is taken from different sources, for example, Blogs, Hotel reviews, and Tweets. In the traditional methods for author profiling task, researchers mainly use features such as words, word classes, and part of speech(POS) n-grams to train their model. In our model, all the tweets are in text and it contains some features. So, we use the LSTM model with some hand-crafted features of Deep Learning to recognize age, gender, and dialect of the authors of a tweet. We performed experiments on the lstm model with and without handcrafted features. We got an improvement in the accuracy, for the lstm-based system with handcrafted features in comparison to the normal lstm based system. Thus it can be said that data shown that the style based features play a crucial role in recognizing the author profiles.

2 Related Work

Author profiling has attracted researchers and other different competitions. Researchers have studied the dependence of linguistic features and the profile of the author. This dependency is a subject of interest for different areas like linguistics, psychology, and natural language processing [6]. Researchers use the syntactic, lexical, and structural features for recognizing the gender and the age group of authors. They used the decision tree for identifying the author profile. This research methodology is also helpful for other applications like security, criminal detection, and author detection [6]. Researchers focused on the representation of the documents, to improve the representation of tweets. They computed high quality discriminative and descriptive features built on the top of the textual features (e.g., content words, function words, punctuation marks, etc.) by exploiting discriminative and descriptive features [1]. Some researchers used typed character n-grams, lexical features, and non-textual features (domain names) for the author profiling task [5]. Researchers also tried deep learning models to directly learn the gender of blog authors. They used Window Recurrent Convolutional Neural Network [2]. Use of Language and Author Profiling uses Computational linguistics approaches, Author Profiling Tasks, Neurology [4].

3 Dataset Description

The dataset was provided by the APDA track organizers of FIRE 2019^4 . It consists of Arabic tweets of 5 set of tweet data. Each data has 100 tweets of 450 users [3]. Since we are using deep learning-based model, so we needed a large

⁴ https://www.autoritas.net/APDA/

set of tweets. We considered each tweet as a sample, thus we have 45000 sample of tweets divided into two class. The users who were male, their tweets were labeled with the male, and similarly for females. Similarly, for age and dialect, 45000 samples were created. Finally, after getting the results, the decision is taken on the majority voting basis. For example, We have 450 tweets of 1 user in 1 document. As per our method, we will get result for all 450 tweets. Let say for 200 of the tweets, the result is Male, and female for remaining then the final result for that document will be female. In this way, we have made the data samples and concluded the results.

These tweets are in Arabic language having Emoticons, emojis, #mentions, @mentions, and URLs. Since the provided training corpus consists of tweets are with HTML format, firstly, we extract all the Tweets from its HTML format to simple text and then applied the preprocessing step for cleaning the tweets. The preprocessing stage is useful as it reduces non-textual features to their semantic classes. We used these preprocessing steps before the extraction of features.

URLs: The URLs are deleted. @mentions: The @ mentions were deleted. Emoticons: Emoticons provide useful style-based information. It informs about the view of a specific user. We captured their presence only. If the emoticon is present, then 1 otherwise 0. Furthermore, we apply the following normalization: Punctuation marks: The punctuation marks are split from adjacent word and captured their presence separately. The Stopwords and Punctuation were also removed from the tweets.

4 Evaluation Framework

In this section, we are discussing the proposed architecture, performance measures and the results.

4.1 Proposed Architecture

We applied a deep learning model for the author profiling task. We have used long short term memory network and applied to the tweets as a classification problem.

Model-I Long short Term Memory(LSTMs) are designed to learn the longterm dependency of text-based problems. LSTMs remember the information for long periods and it is their default behavior⁵. In our LSTM model, firstly, we create a word embedding matrix, a tokenizer and a vocab for our training corpus to determine the unknown words. We randomize our training corpus to make it more efficient. We introduce three layers with different activation keys(relu, sigmoid) to train our training corpus on the LSTM model and test it on the test corpus. We use Binary cross-entropy as a loss function to use for binary decisions and adam as our optimizer in the compiling of our model. In subfigure 1a, we have shown the structure of the model.

⁵ https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Chanchal et al.

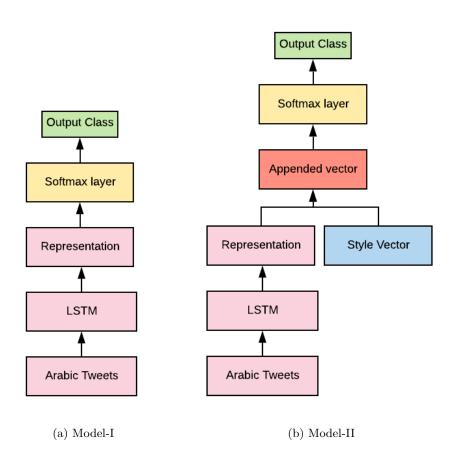


Fig. 1: Proposed Architectures

Model-II We tried a simple variant of the Model-I discussed in 1a, by adding some extra hand-crafted features. We added the hand-crafted features in the developed model to check the performance of the system. In subfigure 4.1, we have shown the structure of the model. Below are the additional handcrafted features:

- Emoji Count: It counts the total number of emoji presents in the tweet.
- Word Count: It counts the total number of words present in the tweet.
- Polarity of Sentiment Analysis: It gives, 1 for positive, -1 for negative and 0 for neutral sentiment.
- Mean Word Length: It calculates the average length of the words present in the tweet. It is the ratio of the total length of words present in the tweet to the total no. of words present in the tweet.
- Sentence length: It gives the length of the sentence in the tweet.

4

- *Special Character*: It gives the total count of punctuation present in the tweet.
- Unrepeated Words: It gives the total number of the words which appeared only once in the tweet.
- URL Extractor: It gives 1 if URL is present in the tweet otherwise it gives 0.

4.2 Evaluation Framework

We used Aravec [8], to create vectors of Arabic words. Word embeddings are the vector of words, which satisfy the semantic property of words [9]. They split the words of a tweet using space and applied a twitter-based embedding to get the vector of words. After the creation of vectors, the tweets are applied to the LSTM layer and then a softmax layer is applied to get the final class of the tweet.

For model-II, we extracted the features from the tweet data and appended the extra features to the output of the LSTM layer. After that, the final result is extracted from the softmax layer. The Github link for the proposed framework is given below. 6

4.3 Results

The performance of the system is evaluated based on the accuracy of the developed system. The Accuracy of a system is the ratio of the total number of instances predicted correctly to the total number of instances present in the data.

$$Accuracy = \frac{P}{P+Q} \tag{1}$$

Where, P is the number of instances predicted correctly, and Q is the number of instances predicted incorrectly.

Data_name	Gender	Age	Dialect
15	82.01	87.58	93.24
16	86.79	82.78	95.04
17	78.27	80.73	90.07
18	85.30	84.58	96.06
19	81.77	84.32	89.86

Table 1: Accuracy of the developed system on training data

 6 https://github.com/chanchalIITP/Author-Profling-FIRE-2019

Data_name	Gender	Age	Variety
15	59.16	68.96	74.35
16	63.24	70.68	77.84
17	58.33	65.14	76.95
18	61.35	71.21	80.04
19	62.05	62.56	76.82

Table 2: Accuracy of the developed system for training data on cross-validation

We found the accuracy of model-I was less than model-II, thus we are reporting the performance of model-II only. In table 1, we have shown the accuracy achieved on training data, and in table 2, we have shown the cross0validation accuracy on the training data. We performed 5-fold cross-validation on the training data to check the performance of the system and reduce the generalization error. It was done because we didn't have the labels of test data before submission.

For training data, model-II is giving better results than model-I. The performance is similar on test data too. The model-II achieved an accuracy of 66.25% for gender, 22.22% for age, and 80.28% for language variety. The joint performance of the system is 0.1083. While the joint accuracy of model-I is 0.0722. In table 3, we have shown the results of the proposed systems on test data.

Table 3: Performance of the proposed system on test data

Model	Gender	Age	Variety	joint
Model-I	57.64	27.50	55.14	0.0722
Model-II	66.25	22.22	80.28	1.083

5 Conclusion and Future work

In this work, we have presented the task of automatic classification of the author's gender, age, and dialect from their writing. This work attracts several potential applications like security, forensics, and marketing, etc. We have developed an lstm-based neural network model for recognizing the age, gender, and language variety of an author by using his/her written text. We have used some style-based features for improving the performance of the lstm-based system. In the future, we will try to optimize the neural network architecture to enhance the efficiency of the system. We would also look into task-specific handcrafted features, to improve the performance of the system. We will work on the using the properties of homonyms in our feature detection.

References

- Alvarez-Carmona, M.A., López-Monroy, A.P., Montes-y Gómez, M., Villasenor-Pineda, L., Jair-Escalante, H.: Inaoes participation at pan15: Author profiling task. Working Notes Papers of the CLEF (2015)
- 2. Bartle, A., Zheng, J.: Gender classification with deep learning. In: Technical report. The Stanford NLP Group. (2015)
- Rangel, F., Rosso, P., Charfi, A., Zaghouani, W., Ghanem, B., Snchez-Junquera, J.: Overview of the track on author profiling and deception detection in arabic. In: Mehta P., Rosso P., Majumder P., Mitra M. (Eds.) Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings. In: CEUR-WS.org, Kolkata, India, December 12-15 (2019)
- Mansanet, J., Albiol, A., Paredes, R.: Local deep neural networks for gender recognition. Pattern Recognition Letters 70, 80–86 (2016)
- Markov, I., Gómez-Adorno, H., Sidorov, G., Gelbukh, A.F.: Adapting cross-genre author profiling to language and corpus. In: CLEF (Working Notes). pp. 947–955 (2016)
- Patra, B.G., Banerjee, S., Das, D., Saikh, T., Bandyopadhyay, S.: Automatic author profiling based on linguistic and stylistic features. Notebook for PAN at CLEF 1179 (2013)
- Rangel Pardo, F.M., Celli, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd author profiling task at pan 2015. In: CLEF 2015 Evaluation Labs and Workshop Working Notes Papers. pp. 1–8 (2015)
- 8. Soliman, A.B., Eissa, K., El-Beltagy, S.R.: Aravec: A set of arabic word embedding models for use in arabic nlp. Procedia Computer Science **117**, 256–265 (2017)
- 9. Word embedding: Word embedding Wikipedia, the free encyclopedia (2019), https://en.wikipedia.org/wiki/Word_embedding, [Online; accessed 16-march-2019]