A CNN based approach to Roman Urdu Author Profiling

Saad Ahmad Khan (20100231) and Soban Ali (20100211)

Abstract: Author profiling has been a widespread focus of research owing to its dynamic applications, but has received little limelight in our local context (Roman Urdu based texts). Various machine learning techniques have been used to solve this issue and we opted to create an efficient convolutional neural network (CNN) based model to solve this problem along with generating word embeddings model for Roman Urdu, a novel approach for Roman Urdu texts. Two embedding models were created and tested. The outputs from the embedding model were fed into the CNN, achieving an accuracy of about 90%.

1. Introduction

User profiling is a task where a user's written text is analyzed to identify the demographic traits such as gender, age, native language, etc. The subject of author profiling is beneficial in many domains such as digital forensic analysis [Glance], marketing intelligence for business [Juola], sentiment analysis and classification for social and physiological behaviors [Anstead]. The profile of an author can be either: (1) monolingual, or (2) multilingual. In the former case, the entire author profile is written in one language, while in the case of the latter, a single author profile will contain text in two or more languages. Author profiling on profile containing text in two or more languages is known as Multilingual Author Profiling [Fatima], which is the focus of our research.

With the advancement of technology and the Internet, people can interact globally via different mediums (text messaging, social media, blogs, etc.). The phenomenon of multilingualism emerged due to the communications among various nationalities having different native languages. People generally use a common language such as English to communicate but are inclined to use their native language, and this fusion creates vocabulary consisting of many different languages.

The languages used on social media platforms tend to differ a lot from the dense vocabulary found in formal literature. Hence, the feature extraction process differs depending on the type of text being used. In previous research studies, different genres (Twitter, Facebook, blogs, web forums) have been considered for mostly English and other European languages in monolingual setting [Burger]. The methodology varied from using a Weka Tool, to supervised and unsupervised machine learning such as random forest and naive bayes. Age prediction task has also been done by using Linear Regression method [Nguyen]. While a few research papers also incorporated deep learning models for instance LSTM, RNN and CNN. These strategies have shown promising results when it comes to identifying the age, gender and personality type. Analysis has also been done on the sentimental analysis of user posts classifying them as positive, negative or neutral. Bayesian networks have been utilized to identify and gauge user interests.

Notable work regarding author-profiling can be seen on PAN challenge. Miura et al. proposed two deep learning based approaches that exploits from both context-based and style-based features by extracting features from both word level and character level information. Their architecture was composed of attention mechanism layers, a max pooling layer and fully connected layers. The places of these layers also play an important role in the overall performance. Kodiyan et al. also used a deep learning approach by implementing a bidirectional RNN with Gated Recurrent Units regarding the same challenge. They added an attention layer on tweet level to learn the most important parts of each tweet and exploited from this information to get information about author.

However, the SMS genre has been neglected for author profiling regardless of its global popularity, ease of use and access. The most likely cause of this negligence is its challenging and time consuming data collection as a standard resource. Moreover, the data collected has privacy and ethical concerns associated with it which further increases the reluctance to perform analysis on SMS. Although, collecting SMS messages for creating a standard evaluation resource is a very challenging task, a few efforts have been made in developing datasets by using SMS messages for various tasks including SMS text normalization [Olivia], linguistic[Treurniet], machine translation systems [Song] and spam detection [Giannella]. Among the existing SMS-based corpora, NUS SMS corpus1 is the largest and most widely used SMS based dataset, which was initially developed to improve the predictive text on mobile devices [Chen].

In the recent years, research has been done on multilingual author profiling. One such investigation is that of the user profiling via tweets in English and Spanish[Byot], particularly for cross genre evaluation. The profiling consisted of age and gender classification. The training sets were taken from tweets while genres for evaluation come from blogs, hotel reviews, other tweets collected in a different time, as well as other social media. Comparisons were done between TF IDF as a baseline and average of word vectors, using a Support Vector Machine algorithm. Results show that using average of word vectors outperforms TF IDF in most cross genre problems for age and gender.

2. Methods

We will be applying a CNN based technique for efficient user profiling from Roman Urdu SMS corpora, based on self trained word2vec embedding and Universal Sentence Encoder (USE) by Google and compare their performances.

2.1 Roman Urdu Corpora

The training and testing corpus will be the publicly available Roman Urdu SMS corpus that has been obtained from Mehwish Fatima. The corpus contains 810 author profiles, wherein each profile consists of an aggregation of SMS messages as a single document of an author, along with seven demographic traits associated with each author profile: gender, age, native language, native city, qualification, occupation and personality type (introvert/extrovert).

The dictionary will contain one more data set:

- Social media comments, with positive and negative labeling. https://github.com/Smat26/Roman-Urdu-Dataset Roman Urdu dictionary, with English definitions

Our work was divided into two main parts, (i) generating embedding and (ii) generating a CNN model, both of which will be discussed in the following paragraphs. Initial steps included cleaning the entire dictionary to make it relevant to the above-mentioned tasks:

- Removing all irrelevant characters such as any non-alphanumeric characters.
- Tokenizing your text by separating it into individual words. Removing words that are not relevant, such as mentions, hashtags or URLs.
- Converting all characters to lowercase so that case has no effect on the model
- Removing emojis

The initial steps resulted in one author profile being discarded due to the data being corrupted, hence, the total number of authors now numbered 809.

2.2 Word Embeddings

Word embedding is an approach to provide a dense vector representation of words that capture something about their meaning. The resultant vector is in a form that enables machine learning algorithms to process natural language as it maps words to real numbers. A simple approach would be to utilize one hot end encoding. However, this will result in a very sparse matrix and will introduce a new dimension for each category.

There are various techniques available to obtain word embedding, word2vec, glove, , etc. However, most of these are only feasible for English, or languages for which word embedding are already available. After the process of elimination, we reached down to word2vec and Universal Sentence Encoder(USE) by Google.

Word2vec gives us the liberty to train our own model, provided tokenized data is available. Therefore, we are going to train word2vec using a roman Urdu vocabulary that was created by us. The vocabulary will contain data from SMS corpus and social media comments. The use of a diverse dictionary will improve the training of our model, considering there are no efficient models available.

USE is a fairly recent sentence encoder that converts a list of texts into an encoding matrix of size 512. Google claims to have incorporated the support of languages other than English but

Urdu is not in the official list. However, we opted to include USE as well to evaluate the performance of our CNN model with a trained and untrained embedding.

The two models are also distinct in the vectors that they create. Word2vec provides us word level embeddings whereas USE creates sentence level embeddings. Thus, this variation will also be considered to compare word and sentence level embeddings.

2.3 Convolutional Neural Network (CNN)

CNN has most of its application in Computer Vision. Just like images can be represented as an array of pixel values (float values), similarly we can represent the text as an array of vectors(each word mapped to a specific vector in a vector space composed of the entire vocabulary) that can be processed with the help of a CNN .When we are working with sequential data, like text, we work with one dimensional convolutions, but the idea and the application stays the same. We still want to pick up on patterns in the sequence which become more complex with each added convolutional layer.

But before applying convolutions over input data we will have to convert all input data into equal length vectors. For this, we found the maximum length text from entire corpus and padded all data points equal to that length. This will avoid the loss of any information and serve the purpose i.e. feature extraction quite well. Our model architecture is composed of an embedding layer, followed by a 1D convolutional layer (128 filters and filter size equal to 3) and 1D global max pooling layer. The outputs of these layers are fed to one vanilla hidden layer and then finally to classification layer. The embedding matrix has been generated earlier so we froze the parameters of embedding layer while CNN model training and it makes the overall trainable parameters to be equal to 100751.

2.4 Model Training

We further divided our analysis into two main approaches. First, we used only a single SMS as an input to our model, trained and tested its accuracy for gender determination. Second, we used a concatenation of all SMS by a particular author as an input to our model and trained and tested its accuracy accordingly. Results of both these approaches will be compared below.

Our initial aim was to generate a multi label classification model, however, the results for data labels other than gender, proved to be of no value, with accuracies barely touching 40%. Hence, we restricted our analysis to gender classification. To avoid the problem of overfitting at the very initial epochs, we will also be using dropout regularization.

3. Results

The SMS corpus consisted of 809 authors and 78117 text messages. The testing dataset was fixed as 20% of the entire dataset. The following paragraphs outline the various results that we generated using our model.

3.1 Individual SMS Gender

Each SMS was individually taken as a single input to the CNN, hence we had 78117 data points, consisting of 19530 testing samples. The embedding model used in this case was word2vec. An accuracy of **78.7%** was achieved.

3.2 SMS Grouped by Author

A second approach was to group the messages according to their authors, resulting in 809 data points, consisting of 162 testing samples. Following results were generated using both our pre-mentioned embedding models.

3.2.1 Word2Vec

As the name suggests, word2vec was set as the embedding layer and the gender was classified. This variant turned out to be the best combination, achieving an accuracy of **90.1%**.

3.2.1 Universal Sentence Encoder (USE)

The sentences were passed through the encoder and an encoding matrix was generated. This matix acted as the embedding layer of the CNN. The output parameter, gender, had to be one hot encoded to match the dimensionality of the output from USE. The model achieved an accuracy of **77.2%**.

3.3 Personality Type and Age

As word2vec provided us with promising results, we expanded our model to classify personality type and age of the author. The personality type resulted in an accuracy of **59.3%** and the latter **36.4%**.

The summary of the results can be seen in table 1.

Result	Embedding	Accuracy(%)
SMS Grouped by Author	**Word2Vec**	**90.1**
	USE	77.2
Individual SMS Gender	Word2Vec	78.7
Personality Type	Word2Vec	59.3
Age	Word2Vec	36.4

Table 1

3. Discussion

As the above comparison shows that instead of using a single text SMS to determine the gender of the author, using multiple text messages by the same author as a single input will give us a better prediction of the author's gender. Moreover, our results also suggests that using this approach, it is not possible to guess other demographic traits of the author i.e. age group, or personality type (introvert/extrovert) etc. The possible reasons of it's inefficiency to predict other demographic traits might be that:

1. There are not enough training samples
2. The data we have does not generalize well on CNN model
3. Missing focus on tweaking the hyperparameters because of low computational power i.e. number of CNN filters, size of filters, dropout and learning rate etc.

Comparing the two embedding models i.e. word2vec and universal sentence encoder, the results show that word2vec performs remarkably better than USE. Word2vec was trained by us on our corpus, whereas USE is pre trained, with Google not disclosing the parameters that would allow us to train it to our needs. Hence, this shows that a trained embedding model is crucial in order to achieve higher levels of accuracy for multilingual datasets.Moreover, it is also apparent that word level embeddings fair better than sentence level embeddings when combined with a CNN mode.

Furthermore, when combined with word2vec, the CNN model achieved an impressive accuracy of 90.1%. This shows a considerable improvement over the previous multilingual author profiling techniques that have been utilized as mentioned in section 1 of this paper.

4. Conclusion

The CNN model combined with the word2vec embedding has shown itself to be a suitable combination for the purposes of Roman urdu author profiling. The model shows promising results, outperforming the existing models, and can be further improved upon to expand the scope of the classification parameters. Furthermore, word level encodings are the viable option when working with deep learning models.

5. Code Files and Trained Models

Use Google Colab to view below mentioned code files.

1. Author Profiling using word2vec
2. Author Profiling using USE
3. Trained word2vec and CNN model

6. References

1. Anstead, N., O'Loughlin, B.: Social Media Analysis and Public Opinion: The 2010 UK General Election. Journal of Computer-Mediated Communication 20(2), 204– 220 (2015)
2. Boyot,R: Multilingual author profiling using word embedding averages and SVMs, IEEE 2015
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 (EMNLP). pp. 1301–1309. Association for Computational Linguistics, Edinburgh, United Kingdom (2011)
4. Chen, T., Kan, M.Y.: Creating a live, public short message service corpus: the NUS SMS corpus. Language Resources and Evaluation 47(2), 299–335 (2013)
5. Fatima, M., Hasan, K., Anwar, S., Nawab, R.M.A.: Multilingual author profiling on Facebook. Information Processing & Management 53(4), 886–904 (2017)
6. FATIMA, M., ANWAR, S., NAVEED, A., ARSHAD, W., NAWAB, R., IQBAL, M., & MASOOD, A. (2018). Multilingual SMS-based author profiling: Data and methods. Natural Language Engineering, 24(5), 695-724. doi:10.1017/S1351324918000244
7. Giannella, C.R., Winder, R., Wilson, B.: (Un/Semi-)supervised SMS text message SPAM detection. Natural Language Engineering 21(4), 553–567 (2015)
8. Glance, N., Hurst, M., Nigam, K., Siegler, M., Stockton, R., Tomokiyo, T.: Deriving Marketing Intelligence from Online Discussion. In: KDD '05: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. pp. 419–428. Chicago, Illinois, USA (2005)
9. Juola, P.: Industrial Uses for Authorship Analysis. In: Mathematics and Computers in Science and Industry, pp. 21–25. INASE (2015)
10. Oliva, J., Serrano, J.I., Del Castillo, M.D., Igesias, A.: an SMS normalization system integrating multiple grammatical resources. Natural Language Engineering 19(01), 121–141 (2013)
11. Song, Z., Strassel, S., Lee, H., Walker, K., Wright, J., Garland, J., Fore, D., Gainor, B., Cabe, P., Thomas, T., Callahan, B., Sawyer, A.: Collecting Natural SMS and Chat Conversations in Multiple Languages: The BOLT Phase 2 Corpus. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik, Iceland (2014)
12. Treurniet, M., De Clercq, O., Van Den Heuvel, H., Oostdijk, N.: Collecting a corpus of Dutch SMS. In: 8th International Conference on Language Resources and Evaluation Conference (LREC 2012). pp. 2268–2273. European Language Resources Association (ELRA), Istanbul, Turkey (2012)
13. Nguyen D, Smith NA, Rose CP (2011) Author age prediction from text using linear regression. In:′ Proceedings of the 5th ACL-HLT workshop on language technology for cultural heritage, social sciences, and humanities. Association for Computational Linguistics, pp 115–123
14. Miura, Y., Taniguchi, T., Taniguchi, M., Ohkuma, T.: Author Profiling with Word+Character Neural Attention Network—Notebook for PAN at CLEF 2017. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Evaluation Labs and

Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland. CEUR-WS.org (Sep 2017), http://ceur-ws.org/Vol-1866/

15. Kodiyan, D., Hardegger, F., Neuhaus, S., Cieliebak, M.: Author profiling with bidirectional rnns using attention with grus. In: CLEF (2017)