Egyptian Informatics Journal 21 (2020) 159–167

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

Gender identification for Egyptian Arabic dialect in twitter using deep learning models

Shereen ElSayed *, Mona Farouk

Cairo University, Faculty of Engineering, Egypt

ARTICLE INFO

Article history:
Received 26 June 2019
Revised 27 October 2019
Accepted 23 April 2020
Available online 21 May 2020

Keywords:
Gender identification
Egyptian Arabic text classification
Deep learning
Natural language processing
Social Media analysis and mining

ABSTRACT

Although the number of Arabic language writers in social media is increasing, the research work targeting Author Profiling (AP) is at the initial development phase. This paper investigates Gender Identification (GI) (male or female) of authors posting Egyptian dialect tweets using Neural Networks (NN) models. Various architectures of NN are explored with extensive parameters' selection such as simple Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long–Short Term Memory (LSTM), Convolutional Bidirectional Long-Short Term Memory (C-Bi-LSTM) and Convolutional Bidirectional Gated Recurrent Units (C-Bi-GRU) NN which is tuned for the GI problem at hand. The best acquired GI accuracy using C-Bi-GRU multichannel model is 91.37%. It is worth noting that the presence of the bidirectional layer as well as the convolutional layer in the NN models has significantly enhanced the GI accuracy.

1. Introduction

The world is evolving now around social media and its significance to our daily life. Although many individuals use social media to connect with people, others use bogus accounts to commit cyber stalking, intimidation and violence. Around 30% of cyber harassers' gender is unknown*. Although author profiling through English text has been extensively investigated in the literature, Arabic-based author profiling in general, and GI in specific, is under studied.

PAN** (a community that investigated text based tasks) started to add Arabic in their GI task dataset in 2017. The approaches investigated in PAN GI tasks were not specific for Arabic language in addition to the low GI accuracies achieved by their proposed solutions. There is a scarcity in the work that targets GI for Egyptian dialect in specific. Previous work in GI investigated several architectures such as CNN, GRU, LSTM, attention layers. . .etc. This work builds a state of the art NN model for the GI in Egyptian dialect. The model aims to achieve higher GI accuracy over multiple Egyptian dialect datasets.

In this paper, the GI problem is investigated on a labelled Egyptian dialect dataset obtained from Twitter. The main research questions considered are: *i)* What is the best NN model that achieves the highest GI accuracy through the Egyptian dialect tweets? *ii)* What is the key component/layer in the NN model for the problem at hand? Multiple NN models were studied with architectural layers that have been previously employed for text classification problems in general, and for the GI problems in specific. Examples of the NN architectures are: *i)* Convolutional Neural Network (CNN), *ii)* Long-Short Term Memory (LSTM), *iii)* Bidirectional Long-Short Term Memory (Bi-LSTM), *iv)* Convolutional Bidirectional Gated Recurrent Unit (C-Bi-GRU), and finally, *v)* a simple NN architecture with fully connected layers. The rest of the paper is organized as follows: Section 2 represents the related work on GI problem considering Arabic language. Next, Section 3 introduces the datasets used in learning and testing phases. Additionally, Section 3 explains in details the conducted neural network models for the GI. Section 4 illustrates the experiment as well as the results for the GI, when EDGAD and PAN AP'17 datasets are considered. Section 5 provides an analysis of the results obtained for the proposed NN models in comparison with the state-of-the-art competitors. Finally, concluding remarks are drawn in Section 6.

* Corresponding author.
E-mail addresses: shereenhussein@rocketmail.com (S. ElSayed), mona_farouk@eng.cu.edu.eg (M. Farouk).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2020.04.001
1110-8665/

2. Related work

Hussein, S. et al. [7] proposed a classical machine learning approach for the GI problem considering Egyptian dialect. The authors worked on extracting gender features from the text as semantic features. Their proposed model considered two feature vectors; *i)* A mixed feature vector combining embedding vector representation and semantic features employed with Random Forest classifier, and *ii)* N-gram feature vector used with Logistic Regression classifier. They achieved 87.6% accuracy for the GI problem over a labeled dataset they retrieved from Twitter for Egyptian dialect tweets.

Bsir, B. and Zrigui, M. [2] used NN model with GRU to identify the gender of authors from Facebook's posts and Twitter. The input to the NN model is preprocessed and split into two: embedding layer and stylometric features extraction phases. The output of the embedding layer is connected to a bidirectional GRU layer then to an activation layer. On the other hand, the stylometric features are normalized and connected directly to the same activation layer used after the GRU layer. They compared their results with the best results by Basile et al. [1] in PAN' AP (2017). The achieved result is less than the best result in PAN'AP (2017).

Estruch et al. [8] enhanced an early fusion model, which was based on performing fusion after the decision level single source classification. The authors achieved 91% GI accuracy on an English dataset in Singapore retrieved from Foursquare, Instagram and Twitter.

Kodiyan et al. [11] investigated gender detection for different languages with multiple dialects, such as English, Spanish, Portuguese and Arabic. They investigated the validity of CNN, and the bidirectional Recurrent Neural Network (RNN) with GRU. It turns out that RNN with GRU outperforms CNN in terms of accuracy. As a result, they achieved 79.03%, 72.57%, 79.5%, and 71.58% GI accuracy for English, Spanish, Portuguese, and Arabic, respectively.

Franco-Salvador et al. [9] used the same dataset considered by Kodiyan et al. [11] to identify the language and author gender as well. The authors used skip-gram model [4] for word embedding exploiting the words' morphology by means of character N-gram embedding. Their architecture was a Deep Average Network (DAN) consisting of multiple hidden layers with Rectified Linear Unit (ReLU) function. They achieved 76.6%, 79.6%, 76.9%, and 77.2% GI accuracy for Arabic, English, Portuguese, and Spanish, languages respectively. The approach of N-gram embedding underperformed when considered in conjunction with word embedding model.

Miura et al. [13] proposed a model for word and character embedding. They considered RNN for the word embedding case, while CNN for character embedding case. Additionally, the authors added multiple attention layers. The embedding layer was initialized by a model that was trained through Twitter dataset. For the GI task, the average achieved accuracy for dialectical Arabic was 79.17%.

Deep learning usually required a massive amount of data for training. As a result, Veenhoven et al. [22] translated the gold standard data to the language of interest. Additionally, using PAN-AP' 18 dataset the authors used Bi-LSTM and CNN architectures to solve the GI problem. However, when the RNN was considered, the highest GI accuracy obtained was 79.3%, 80.4%, and 74.9%, for English, Spanish and Arabic languages, respectively.

Ranjan, S. et al. [17] proposed an approach for identifying fake user accounts in Twitter. They used multiple features, such as postings (text, image, video...etc.), followed accounts, followers' accounts, replies, retweets, media contents, tagging, likes, direct messages and URLs. They used graph-based and machine learning methods to identify the fake accounts by extracting their features.

3. Materials and methods

3.1. Datasets

The dataset used in this work consists of Egyptian dialect tweets. There are two datasets that have been considered, Egyptian Dialect Gender Annotated Dataset (EDGAD) from [7] and PAN (part of CLEF – conference of labs and evaluation forum) Author Profiling task 2017 (PAN AP'17).

3.1.1. Edgad

EDGAD has been compiled by [7] for the purpose of research on gender identification problem using Egyptian dialect. The dataset consists of Egyptian dialect tweets collected from 70 public Twitter accounts per gender. The public accounts included active social media influencers and famous people with native Egyptian dialect. For each account, 1000 tweets were collected. The tweets of each account are stored in a separate file with a unique identifier. Some of the balanced EDGAD statistics are represented in Table 1. For more details on the dataset, the interested readers can refer to [7].

3.1.2. Pan AP'17 dataset

The PAN Author Profiling in 2017 (PAN AP'17) dataset was compiled by [15] for the purpose of author profiling considering multiple languages with different dialects. This dataset consists of 4 Arabic dialects tweets: Egyptian, Gulf, Levantine and Moroccan. There are 600 authors for each dialect, with 100 tweets per author. In this paper, the authors with Egyptian dialect tweets will only be considered.

3.2. Deep learning models

Deep learning has proven its success in multiple Natural Language Processing (NLP) problems with different models and architectures. In [3] the authors proposed NN-based models to solve the GI problem. The models were designed based on a mixture of building units that achieved good performance in several text classification tasks. For the GI problem, several architectures such as CNN, LSTM have been studied in the literature and have shown great promise in solving the problem at hand. This work first starts by the simple NN models and utilizes them to solve the GI problem. Afterwards, a level of sophistication is added to the NN models by combining different layers from several NN models in order to achieve higher GI accuracy. In this section, a set of NN architectures that were investigated in this work will be presented. For each NN architecture, the parameters have been fine-tuned to optimize the GI accuracy.

The optimal parameters were identified for the NN architectures by conducting experiments with different number of layers, number of nodes per layer, activation functions, DropOut (DO) probability and arrangements of layers.

In the following subsections, first, a brief illustration about the layers types is given. Afterwards, various NN models are built by employing these layers.

Table 1
EDGAD Statistics from Hussein, S. et al. [7].

Metric	Female	Male
Number of Accounts	70	70
Number of tweets	1000	1000
Average tweet length	14	14
Vocabulary per gender	9,687	10,233

3.2.1. Layers

3.2.1.1. Embedding layer. The embedding layer creates a representation of the words in a vector form with a fixed dimension. Miyato et al. [14] discussed the usage of the embedding layer in comparison with one-hot vector representation. It turned out, that the embedding layer achieved improvements in the word representation and created a model which is less prone to over-fitting in the learning phase. An embedding layer was added to all the models as the first layer, where the vector dimension is set to 100. The input to this embedding layer is a single tweet transformed to a vector of word-identifiers. These word-identifiers are obtained by compiling a vocabulary dictionary, where each unique word is given a unique number. The output of this embedding layer, for each input tweet, is a two-dimensional matrix, where each word is represented by the corresponding embedding vector. The embedding layer operation is depicted in Fig. 1. The embedding layer is followed by a drop-out with probability of 0.15. The drop-out was shown to make the network less sensitive to specific weights of neurons. This, in turn, results in a network that is capable of a better generalization and is less prone to over-fitting [21]. An additional flattening layer is added after the embedding layer. The flattening layer is used to reshape the two-dimensional input matrix into a single vector. In Fig. 2, the embedding vectors of each word are concatenated to form a single vector representing the input tweet.

3.2.1.2. Convolutional 1-D layer. For every input tweet, the input to the convolutional layer is the output of the flatten layer which is a vector of length equals to the maximum number words in a tweet multiplied by the embedding vector size. Since the filter window is (4×1), the convolutional layer used is 1 dimension (1D). Hence, the convolutional layer can operate on each input tweet independently. The number of filters considered is 516 filters, and the window size varies from 3 to 6 words. These parameters are fine-tuned for Arabic language expressions, where the words correlation drops significantly beyond the considered window sizes range. Several activation functions have been investigated; it turns out that tanh provides the best performance in terms of GI accuracy.

3.2.1.3. Max-Pooling layer. The max-pooling layer is required to down-sample the input representation, i.e., decreases its dimensionality and prevents over-fitting as has been reported by Collobert et al. [6]. The operation of this layer is illustrated in Fig. 3. The pool size range is chosen from 2 to 4. The pool size range is identified based on several experiments considering 2, 4, 6 and 8 pool sizes. The results show that no performance difference is observed when the pool size is changed from 4 to either 6 or 8. This means that using 6 or 8 pool sizes will add unnecessary redundancy. This layer is followed by a dropout layer with probability of 0.15.

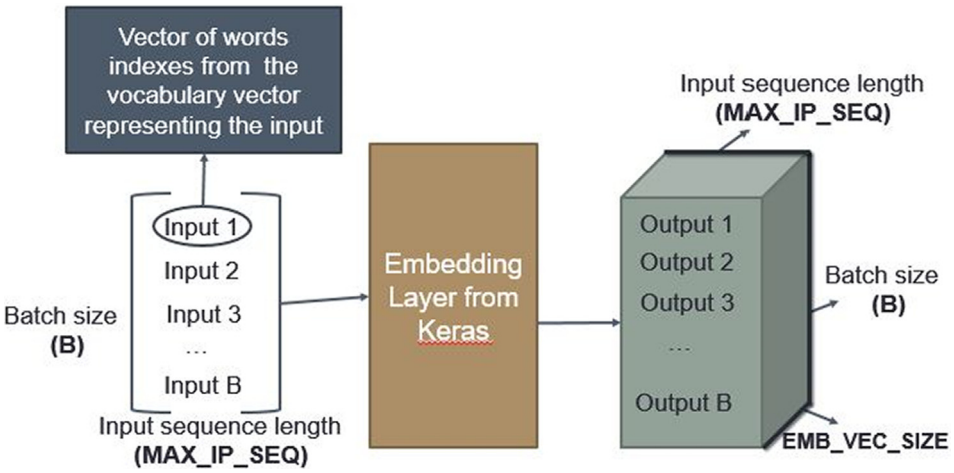

Fig. 1. Embedding layer illustration

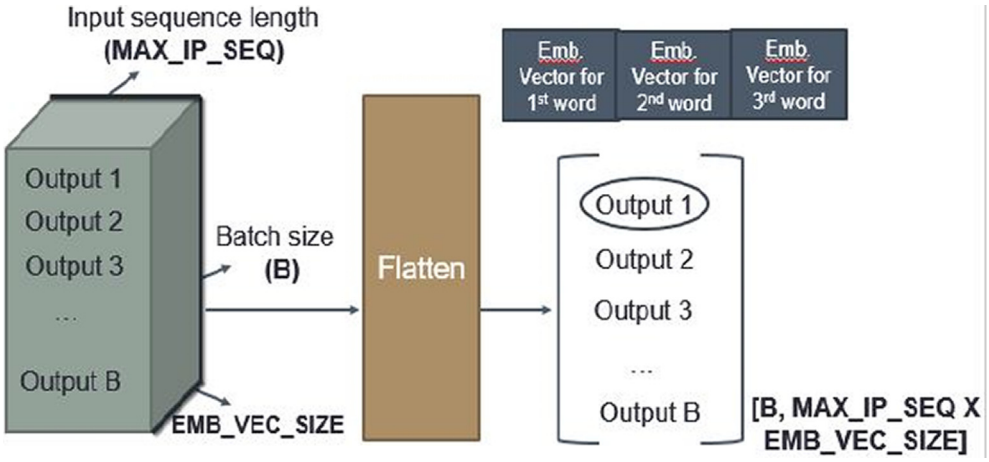

Fig. 2. Flattening layer illustration

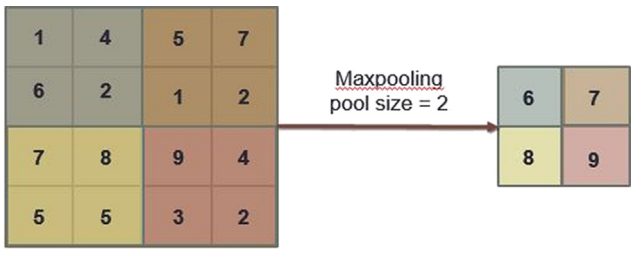

Fig. 3. Maxpool functionality illustration.

3.2.1.4. Lstm. LSTM succeeds in keeping the contextual information of the inputs through different steps. It is widely used in text classification problems, where the premise here is that the current word is highly dependent on the previous ones. LSTMs networks overcomes two main problems in neural networks' learning, the vanishing gradients and exploding gradients. This layer consists of recurrently connecting blocks. Each block consists of three gates, an input, an output and a forget gate. The main purpose of these gates is to read, write and reset the memory block. The used recurrent activation for the gates is Sigmoid and the activation function for the states (hidden and output states) is Scaled Exponential Linear Unit (SELU).

3.2.1.5. Gru. The GRU is a modified version of the LSTM layer. It was initially introduced by [5]. It consists of two gates, a reset gate, and an update gate. Intuitively, the reset gate determines how to combine the new input with the previous memory, and the update gate defines how much of the previous memory to keep around. The used recurrent activation for the gates is Sigmoid and the activation function for the states (hidden and output states) is Scaled Exponential Linear Unit (SELU).

3.2.2. Proposed Deep learning models

3.2.2.1. Model 1 - simple Artificial Neural Network (ANN). This is a model of a simple NN architecture that consists of fully connected layers without the recourse to complex layers such as convolutional or LSTM layers. It consists of an embedding layer as described in Section 3.2.1.1, followed by four fully connected layers containing 80, 40, 30, and 1 node(s), respectively. A DO was added with 0.15 dropping probability. The activation function for all the fully connected layers is ReLU, except for the last layer where the sigmoid function is used instead. The simple ANN is depicted in Fig. 4. For all of the NN architectures, the learning phase is divided into batches, where in each batch, 800 tweets are entered.

3.2.2.2. Model 2 - CNN. In this model the embedding layer is followed by a convolutional 1D layer. The window size is 3, the number of filters is 516, and the activation function is tanh. Afterwards, a max pooling layer with pool size 4 is added. Next, a set of two dense layers containing 30 nodes, and ReLU activation function are added. Additionally, a DO operation is performed with 0.15 dropping probability. The last layer is a dense layer with one node and sigmoid activation function. The overall layers diagram is depicted in Fig. 5.

3.2.2.3. Model 3 - Multichannel CNN. This model consists of three parallel group of layers, also called channels, having different layer parameters as shown in Fig. 6. Each group includes an embedding layer, convolutional layer, and max-pooling layer. The number of filters in the convolutional layers is 516. The window sizes are chosen as 2, 4 and 8 for each of the three parallel groups, respectively. The activation function is tanh and the DO operation is performed with 0.15 probability. The pool size considered for the max pooling layer is 4. The outputs of these channels are concatenated and fed into two dense layers with 30 nodes and ReLU activation function. The last layer is a dense layer with one node and sigmoid activation function.

3.2.2.4. Model 4 - Recurrent NN – Single channel LSTM. As shown in Fig. 7, the first layer is the embedding layer followed by two LSTM layers. Each of the LSTM layers includes 70 nodes and tanh activation function. Next, a dense layer with 100 nodes is added. Afterwards, another dense layer with 70 nodes is placed. The last layer is a dense layer with one node and sigmoid activation function.

3.2.2.5. Model 5 - convolutional bidirectional LSTM NN. This model consists of a single channel including the embedding layer, convolutional 1D layer with window size of 6, and finally a max pooling layer with pool size of 4 as shown in Fig. 8. Next, a bidirectional

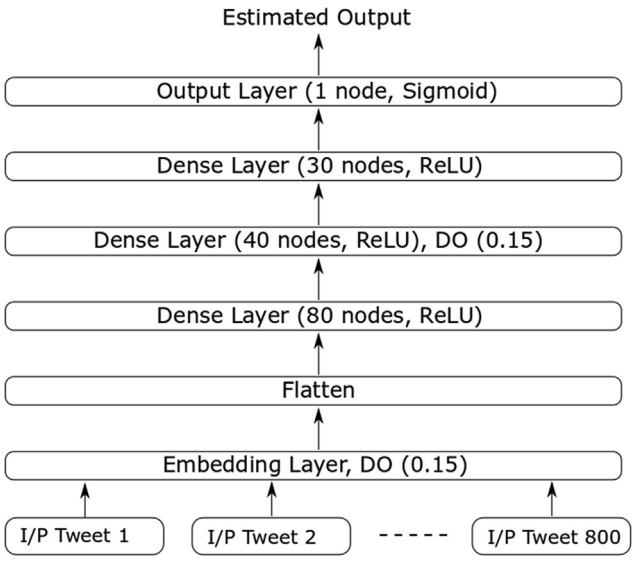

Fig. 4. Model 1- The simple ANN.

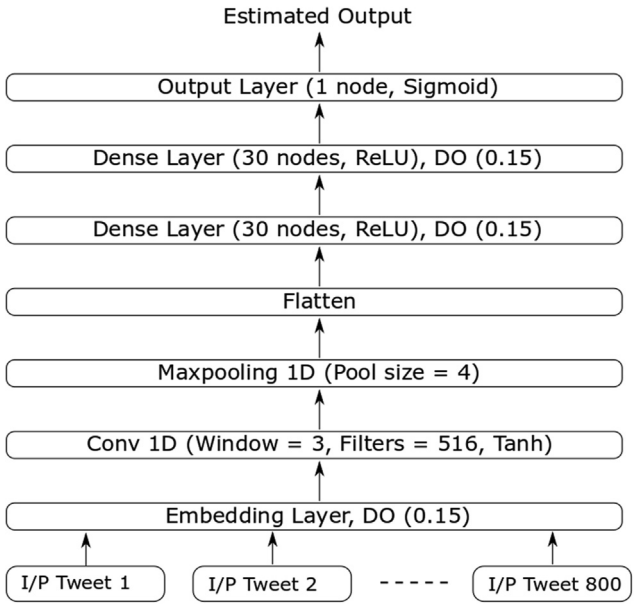

Fig. 5. Model 2- CNN.

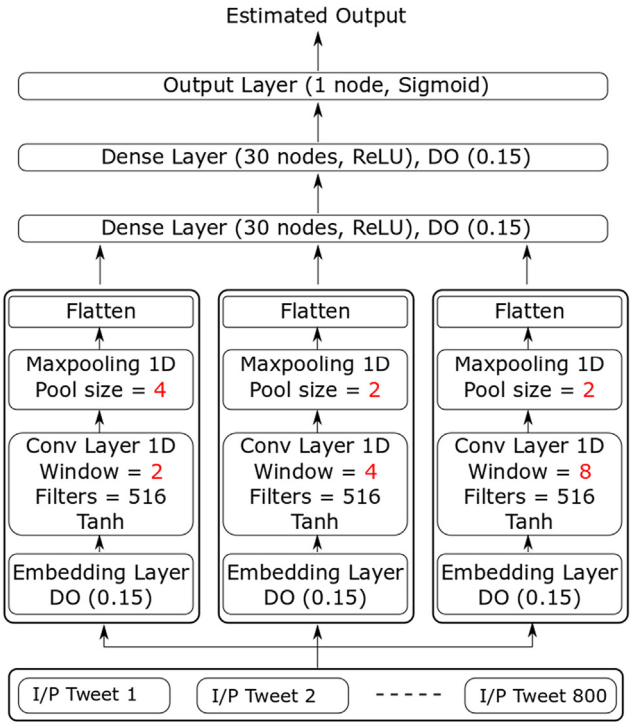

Fig. 6. Model 3- Multichannel CNN.

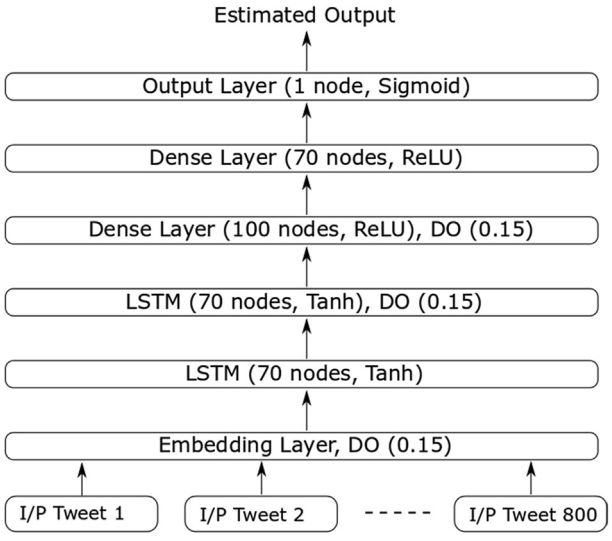

Fig. 7. Model 4- LSTM NN.

LSTM layer is added with 40 nodes. Afterwards, two dense layers are added consisting of 40 nodes and employing the ReLU activation function. The output layer for this model is similar to the one illustrated in the previous models.

3.2.2.6. Model 6 - Multichannel convolutional bidirectional GRU NN. This model consists of three channels as shown in Fig. 9. Each channel starts with an embedding layer, followed by a convolutional layer with 32 filters, DO operation and max-pooling of size 2. The kernel sizes for the three channels are 2, 4 and 8, respectively. The values for the pool size and the kernel sizes are realized after several experiments to achieve the highest GI accuracy for the model. After that, for each channel, there is a bidirectional GRU layer with 40 nodes. Next, the output of the three channels is concatenated and fed to a set of two dense layers with 30, and 20 nodes respectively. The dense layers employed the ReLU activation function. The output layer for this model is similar to the one illustrated in the previous models.

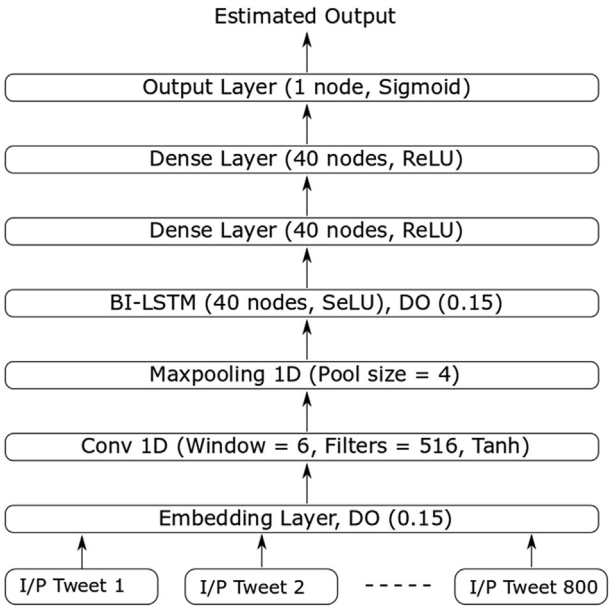

Fig. 8. Model 5 - C-Bi-LSTM NN.

4. Results

In this section, the experiments using the explained models in the previous section are conducted and analyzed. First, the

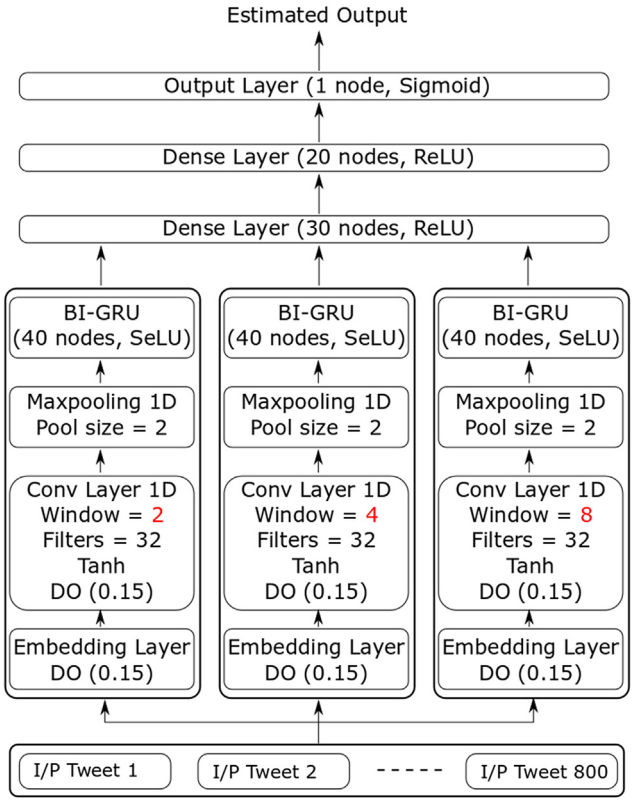

Fig. 9. Model 6 - Multichannel C-Bi-GRU NN.

different models are investigated with various parameters. The tuned parameters are the embedding layer vector size, number of layers, number of nodes per layer, and the activation function per layer. Afterwards, we investigated the types of the NN layers. The values of the parameters used in all the models are optimized to achieve the highest GI accuracy by conducting a wide range of experiments. In the next subsections, the investigations conducted over the hyper-parameters tuning will be illustrated. Additionally, the performance of the NN models is evaluated using multiple input sizes on both EDGAD and PAN AP'17 Egyptian dialect dataset.

4.1. Hyper-Parameters tuning

4.1.1. Embedding vector size

We consider a single channel CNN model with a set of four concatenated tweets as input. The maximum tweet length is 50 characters. The experiments are carried on EDGAD. The results are shown in Table 2, note that using vector size of 100 for words representation has led to the best performance. Additionally, it is worth noting that this conclusion is extendable regardless of the NN model.

4.1.2. Drop-Out

The DO probability used in all the models is 0.15. This probability is achieved after investigating the range [0.05, 0.5]. The experiments are held on the ANN model while fixing all the parameters i.e. number of layers, number of nodes per layer and activation functions. Additionally, the input tweet is a single tweet with maximum length of 50 characters. The GI accuracy with DO probabilities 0.05 and 0.1 is in range 70.1% +/-0.3%. In the experiment with the DO probabilities in the range [0.2, 0.5], the GI accuracy dropped to 68.3% +/-0.5%. The highest GI accuracy achieved using the ANN model with DO probability 0.15 is 71.05% +/-0.3% while fixing all the other parameters.

4.1.3. Convolutional layer

The window sizes for the convolutional layer and the max-pooling are investigated over the CNN model. For the convolutional layer, the window size is in the range of [2,8]. Higher values of window size are investigated, but the GI accuracy did not show any significant change. The filter sizes are experimented in the range [12, 1024], it is noticed that small values (from 12 to 100) for the filters are not enhancing the accuracy. However, higher values (from 500 to 1024) achieved a noticeable enhancement in the GI accuracy on average 2%. The final value for the filter size is chosen as 516 as it was proved by experiments that higher numbers of filters don't cause increase in the GI accuracy.

4.2. Models evaluation

4.2.1. Edgad

First, preprocessing for input data is carried out as follows:

- The non-Arabic characters (English characters, digits, special characters and links) are removed.
- A whitespace between emojis is added to help the tokenizer in separating and extracting emojis types and counts.
- Tweets less than five words are discarded along with the duplicate tweets.

Table 2
Embedding vector size versus the GI accuracy on EDGAD using single channel CNN, 4 concatenated tweets and 50 maximum length.

Vector Size	**100**	150	200	250	300
Accuracy	**82.24**	82.12	82.11	82.05	81.70

Table 3
Model 1 - ANN accuracies for GI on EDGAD using multiple input lengths and number of concatenated tweets.

Maximum Length	Single tweet	4 tweets	8 tweets	12 tweets
50	71.05	80	77.4	77.6
80	71.11	82.43	83.97	76.45
100	71.2	83.21	86.89	83.62
120	71.22	82.87	87.22	86.76
140	71.27	83.5	87.05	**88.11**

Table 4
Model 2 - CNN accuracies for GI on EDGAD using multiple input lengths and number of concatenated tweets.

Maximum Length	Single tweet	4 tweets	8 tweets	12 tweets
50	72.42	82.1	81.66	81
80	72.2	82.64	86.96	85.23
100	72.7	82.9	88.41	88.3
120	72.44	82.7	88.3	88.35
140	72.62	82.8	88.5	**88.6**

Table 5
Model 3 – Multi-channel CNN accuracies for GI on EDGAD using multiple input lengths and number of concatenated tweets.

Maximum Length	Single tweet	4 tweets	8 tweets	12 tweets
50	70.89	82.11	83.2	80.22
80	70.24	83.16	87.06	85.77
100	70.4	83.35	88.67	88.11
120	70.3	83.4	89.13	89.57
140	70.67	83.3	89.32	**91.31**

Table 6
Model 4 - LSTM NN accuracies for GI on EDGAD using multiple input lengths and number of concatenated tweets.

Maximum Length	Single tweet	4 tweets	8 tweets	12 tweets
50	53.17	53.3	53.06	52.1
80	53.13	53.4	53.1	52.5
100	53.12	53.42	53.2	52.6
120	53.08	53.44	53.09	52.62
140	53.11	53.39	53.16	**53.5**

Table 7
Model 5 - Convolutional Bidirectional LSTM NN accuracies for GI on EDGAD using multiple input lengths and number of concatenated tweets.

Maximum Length	Single tweet	4 tweets	8 tweets	12 tweets
50	71.09	81.19	80.57	77.96
80	71.12	82.56	85.8	84.88
100	71.11	82.66	88.38	86.75
120	71.0	82.68	89.15	89.74
140	71.13	82.55	89.5	**90.86**

Table 8
Model 6 - Multichannel Convolutional Bidirectional GRU NN accuracies for GI on EDGAD using multiple input lengths and number of concatenated tweets.

Maximum Length	Single tweet	4 tweets	8 tweets	12 tweets
50	71.33	81.98	81.39	81.2
80	71.35	83.71	87.16	85.14
100	71.29	83.99	88.43	88.45
120	71.31	83.87	89.11	89.46
140	71.3	83.95	89.17	**91.37**

Several experiments are carried out for the NN models in Section 3.2.2 using different number of concatenated input tweets (single, 4, 8 and 12) as input. Additionally, the maximum number of characters in the set of concatenated tweets belongs to (50, 80, 100, 120 and 140). For all the models, Adam optimizer [10] and binary cross-entropy loss function are used. Additionally, the training phase consists of 8 epochs. The validation methodology used is cross-validation with K = 7, where the accounts used in learning are not used in testing. The GI accuracy results for the six models are represented in Tables 3, 4, 5, 6, 7 and 8. Table 9.

4.2.2. PAN AP'17 Egyptian dataset

Similar to EDGAD, the same preprocessing steps are applied on the PAN AP'17 dataset. The experiments results are represented in Table 8. Note that model 6, the multichannel convolutional bidirectional GRU NN, outperforms the other competitor NN models.

5. Discussion

By analyzing the results of the experiments performed using different NN models, different number of concatenated tweets, different maximum characters length, and the following observations are drawn: *i)* Since our problem is a text classification problem, LSTM was expected to achieve high accuracy. However, low GI accuracy is achieved using LSTM alone as in model 4. The reason for the low accuracy is that LSTM requires a large amount of data in text classification problems in general which is not fulfilled with EDGAD or PAN datasets. *ii)* The importance of the bi-direction is emphasized when added to the LSTM, the GI accuracy is dramatically improved from model 4 to 5 by 37.3%. *iii)* The convolutional layer for text classification is proven to be extremely effective in the GI problem as achieved by model 2, where CNN achieved 88.6% for 12 concatenated tweets and 140 input length. *iv)* The multichannel models (3 and 6) show that combining the output of parallel convolutional layers with different window sizes enhance the GI accuracy. *v)* The GI accuracy is slightly improved by adding GRU layer to the existing multichannel CNN model, i.e., model 6. *vi)* Finally, the multichannel convolutional bidirectional GRU model achieves the highest accuracy in GI using EDGAD, which is 91.37% for 12 concatenated tweets and 140 input length.

The best model's accuracy was compared with three competing approaches. *i)* [7] used the EDGAD, they used classical machine learning approaches to achieve 87.6% accuracy, where the proposed deep learning model (Multichannel C-Bi-GRU NN) surpassed it by 3.77% as illustrated in table 10. *ii)* The GI accuracy in PAN AP'17 using their dialectical Arabic dataset is 80.06% and 0.82% by classical machine learning and deep learning respectively. The GI accuracy achieved using Multichannel C-Bi-GRU model on the Egyptian dialect subset of their dataset is 83.7%. *iii)* The models proposed by Sboev et al. [18] are also investigated, their first model which was based on convolutional layers, achieved 68.6% in GI on EDGAD. The second model [18], which was based on bidirectional LSTM layers with convolutional layers in between, achieved GI accuracy of 74.3% on EDGAD.

Other approaches were investigated such as *i)* Schaetti [19] and Franco-Salvador et al. [9] who used character embeddings, character embedding is investigated with the proposed CNN model and achieved GI accuracy less than the word embedding accuracy by 4.2% on average. Word embedding in Arabic outperformed the character embedding in GI. *ii)* Kodiyan et al. [11] experimented the bidirectional RNN with GRU combined with attention mechanism. However, during the design process of the proposed models, the GRU was experimented with the attention mechanism, it achieved lower accuracy than the Multichannel C-Bi-GRU NN (model 6). *iii)* Sezerer et al. [20] used the attention layer with a CNN model. The attention layer did not achieve better GI accuracy when investigated with the models proposed by this work. *iv)* Veenhoven et al. [22] worked on translated data across different languages. The concept of using translated data could introduce noise to the system, as well as incorrect data translation across different dialects. This methodology cannot be used with dialectical languages. However, it helps in systems that work with MSA.

Table 9
GI percentage accuracies for all the models on PAN AP'17 Egyptian dialect using different number of concatenated tweets (single, 4, 8 and 12) with input size 140.

Model	Single	4	8	12
ANN	61.0	72.0	77.4	80.23
CNN	56.3	72.42	80.8	82.2
Multichannel CNN	65.4	71.05	79.87	77.4
LSTM NN	52.3	53.1	53.11	53.4
C-Bi-LSTM NN	57.19	70.4	77.7	79.01
Multichannel C-Bi-GRU NN	65.1	72.4	81.3	**83.7**

Table 10
Comparison between the proposed C-Bi-GRU NN model (model 6) with the best GI accuracy obtained by Hussein, S. et al., [7] and Basile et al., [1] on EDGAD and PAN'AP 2017 Arabic datasets.

	EDGAD	PAN'AP 2017
Hussein, S. et. al. [7]	87.6%	N/A
Basile et al., [1]	N/A	0.82%
Our C-Bi-GRU NN (model 6)	91.37%	83.7%.

The tweets and their corresponding gender classification are analyzed. It was observed that tweets that are very obvious for human to belong to a certain class are mostly correctly classified by all the proposed models. An example of such tweets is "ضوافري لأول مره اسيبها تطول و مبسوطه بيها و هي بتكبر قدامي كده خالص" (For the first time I leave my nails to grow and I am so happy to see it growing in front of me) which is correctly classified as female by all the models. On the other hand, tweets that seem vague for human that he can't correctly identify the gender of their author are mostly misclassified by the models but can be exclusively classified by the C-Bi-GRU. An example of such tweets is "اليوم الي بيبقي فيه سمك ف البيت ده بيبقي من أسوأ ايام حياتي" (The day where there is fish in the home, is one of the worst days in my life). This tweet is correctly classified as female by C-Bi-GRU model.

The time of running the models increases as the model's complexity increases. As an example for this, the C-Bi-GRU and the C-Bi-LSTM models took around one hour to run over the Google Colab's GPU for the single tweets experiments. However, there is another factor for the time complexity, as the number of inputs decrease, but their size increase due to concatenation of tweets together, the time required to run the experiment almost decreased to 15 min.

Since the work in this field is still in its initial phases, the experiments can be extended to investigate the images posted by each account. The input vector for the NN models can be extended to include language specific features for GI i.e. usage of female suffixes and prefixes.

6. Conclusion

In this paper, a solution for the GI problem is proposed and experimented using an Egyptian dialect labelled dataset retrieved from Twitter. Six NN architectures are built and evaluated in terms of GI accuracy. The addition of the convolutional layer and the bidirectional LSTM/GRU layer has significantly improved the GI accuracy. Finally, the C-Bi-GRU model achieves the highest performance for EDGAD and PAN'AP 17, where the GI accuracies are up to 91.37% and 83.7%, respectively.

7. Future work

There are many directions for expanding the work in this open research topic which include:

- Creating a generic solution for gender identification for all the Arabic dialects.
- Using the posted images in the detection of the author's gender as experimented by
- Expanding the work to more author related tasks, such as author's age, identification and obfuscation.

References

[1] Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H. and Nissim. M., N-GrAM: New Groningen Author-profiling Model—Notebook for PAN at CLEF 2017. In Linda Cappellato, Nicola Ferro, Lorraine Goeuriot, and Thomas Mandl, editors, CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland, September; 2017. CEUR-WS.org. ISSN 1613-0073.

[2] Basir, B., Zrigui, M., Enhancing Deep Learning Gender Identification with Gated Recurrent Units Architecture in Social Text. In Computación y Sistemas (An international journal for computing science and applications). Vol. 22, No. 3; 2018, pp. 757–766. doi: 10.13053/CyS-22-3-3036. Available from: http://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/3036.

[3] Belinkov, Y., Glass, J., A Character-level Convolutional Neural Network for Distinguishing Similar Languages and Dialects. In: The Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial). Osaka, Japan, December; 2016. Available from: https://arxiv.org/abs/1609.07568.

[4] Bojanowski, P., Grave, E. and Joulin, A. and Mikolov, T., Enriching Word Vectors with Subword Information. In: arXiv: 1607.04606; 2016. Available from: https://arxiv.org/abs/1607.04606.

[5] Cho, K. Merriënboer, B., Gulcehre, C., Bougares, F., Schwenk, H., Bahdanau, D., Bengio, Y., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In: Empirical Methods in Natural Language Processing. Doha, Qatar, October 25-29, pp. 1724-1734. Available from: https://arxiv.org/abs/1406.1078.

[6] Collobert, R. Weston, J. Bottou, L. Karlen, M. Kavukcuoglu, K. Kuksa, P., Natural Language Processing (Almost) from Scratch. J Mach Learn Res 2011; 12:1532-4435, pp: 2493-2537. Available from: http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf.

[7] Hussein, S., Farouk, M. Hemayed, E., Gender identification of egyptian dialect in twitter. Egypt Informat J; 2019;20:109–116. Doi: 10.1016/j.eij.2018.12.002.

[8] Estruch, C. P., Paredes, R. Rosso, P., Learning Multimodal Gender Profile using Neural Networks. Rec Adv Nat Language Process 2017; Varna: Bulgaria, pp: 577-582. Available from: http://users.dsic.upv.es/~prosso/resources/PerezEtAl_RANLP17.pdf.

[9] Franco-Salvador, M. Plotnikova, N. Pawar, N. Benajiba, Y., Subword-based Deep Averaging Networks for Author Profiling in Social Media. In: Cappellato L., Ferro N., Nie J.L., Soulier L. (Eds.) CLEF 2017 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org; 2017: 1866. Available from: http://ceur-ws.org/Vol-1866/paper_192.pdf

[10] Kingma, D., Ba, J., 2014. Adam: {A} Method for Stochastic Optimization. In: International Conference on Learning Representations. Available from: http://arxiv.org/abs/1412.6980.

[11] D. Kodiyan F. Hardegger S. Neuhaus M. Cieliebak. Author Profiling with Bidirectional RNNs using Attention with GRUs L. Cappellato N. Ferro J.L. Nie L. Soulier CLEF; 2017 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org, vol. 1866. Available from: http://ceur-ws.org/Vol-1866/paper_77.pdf.

[13] Miura, Y., Taniguchi, T., Taniguchi, M., Ohkuma, T., Author Profiling with Word +Character Neural Attention Network In: Cappellato L., Ferro N., Nie J.L., Soulier L. (Eds.) CLEF 2017 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings; 2017. CEUR-WS.org, vol. 1866. Available from: http://ceur-ws.org/Vol-1866/.

[14] Miyato, T. Dai, A., Goodfellow, I., Adversarial Training Methods for Semi-Supervised Text Classification. In: 5th International Conference on Learning Representations, arXiv: 1607.04606. 2017; Toulon: France. Available from: https://arxiv.org/abs/1605.07725.

[15] 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: Cappellato L., Ferro N., Nie J.L., Soulier L. (Eds.) CLEF 2017 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org; 2017. Vol. 1866. Available from: http://ceur-ws.org/Vol-1866/.

[17] Sahoo SR, Gupta BB. Hybrid approach for detection of malicious profiles in twitter. Comput Electr Eng 2019;76:65–81. doi: https://doi.org/10.1016/j.compeleceng.2019.03.003.
[18] Sboev, A. Moloshnikov, I. Gudovskikh, D. Selivanov, A. Rybka, R. Litvinova, T. Automatic Gender Identification of Author of Russian Text by Machine Learning and Neural Net Algorithms in Case of Gender Deception. In: 8th Annual International Conference on Biologically Inspired Cognitive Architectures. 2018; 123: 417-423. Available from: https://www.sciencedirect.com/science/article/pii/S1877050918300656.
[19] Schaetti, N., Character-based Convolutional Neural Network and ResNet18 for Twitter Author Profiling. In: Cappellato L., Ferro N., Nie J.L., Soulier L. (Eds.) CLEF 2018 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org; 2018: vol. 2125. Available from: http://ceur-ws.org/Vol-2125/paper_100.pdf.
[20] Sezerer, E. Polatbilek, O. Sevgili, O. Tekir, S. Gender Prediction From Tweets With Convolutional Neural Networks. In: Cappellato L., Ferro N., Nie J.L., Soulier L. (Eds.) CLEF 2018 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org; 2018: vol. 2125. Available from: http://ceur-ws.org/Vol-2125/paper_116.pdf.
[21] N. Srivastava G. Hinton A. Krizhevsky I. Sutskever R. Salakhutdinov Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J Mach Learn Res 2014; 15: 1929-1958 Available from http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf.
[22] Veenhoven, R. Snijders, S. Hall, G. Noord, R. Using Translated Data to Improve Deep Learning Author Profiling Models. In: Cappellato L., Ferro N., Nie J.L., Soulier L. (Eds.) CLEF 2018 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org; 2018: vol. 2125. Available from: http://ceur-ws.org/Vol-2125/paper_178.pdf.

Further reading

[12] Lotan, G. Graeff, E. Ananny, M. Gaffney, D. Pearce, I., The Arab Spring the revolutions were tweeted: Information flows during the 2011 Tunisian and Egyptian revolutions Int J Commun; 2017. Available fromhttps://www.researchgate.net/publication/235354320_The_Revolutions_Were_Tweeted_Information_Flows_During_the_2011_Tunisian_and_Egyptian_Revolutions.
[16] Rangel F., Rosso P., Montes-y-Gmez M., Potthast M., Stein B., Overview of the 6th Author Profiling Task at PAN 2018: Multimodal Gender Identification in Twitter. In: Cappellato L., Ferro N., Nie J.L., Soulier L. (Eds.) CLEF 2017 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings. CEUR-WS.org; 2018: vol. 2125. Available from: http://ceur-ws.org/Vol-2125/.