2022 30th International Conference on Electrical Engineering (ICEE)

Job Title Prediction from Tweets Using Word Embedding and Deep Neural Networks

Shayan Vassef
*School of Electrical and Computer Engineering
University of Tehran
Tehran, Iran
sh.vassef@ut.ac.ir
†Adak Vira Iranian Rahjo company
Tehran, Iran

Ramin Toosi
School of Electrical and Computer Engineering
University of Tehran
Tehran, Iran
r.toosi@ut.ac.ir
†Adak Vira Iranian Rahjo company
Tehran, Iran

Mohammad Ali Akhaee
School of Electrical and Computer Engineering
University of Tehran
Tehran, Iran
akhaee@ut.ac.ir

***Abstract*—The more social media take its place in our lives; the more critical their analysis becomes and the more researchers' attention is drawn to it. Studies contain various topics such as sentiment analysis, trend prediction, bot detection, Etc. Here, for the first time, we propose a novel method to predict the job title of social media users. Twitter, a popular social media, is our target social media. We introduce a dataset consisting of 1314 samples, including users' tweets and bios. The user's job title is found using Wikipedia crawling. The challenge of multiple job titles per user is handled using a semantic word embedding and clustering method. Then, a job prediction method is introduced based on a deep neural network and TF-IDF word embedding. We also use hashtags and emojis in the tweets for job prediction. Results show that the job title of users in Twitter could be well predicted with 54% accuracy in nine categories.**

***Index Terms*—text classification, social media analysis, Job title prediction, Twitter analysis**

I. INTRODUCTION

Today, social media is an integral part of our daily lives. People spend hours on social media for information, entertainment, or business purposes during the day. Therefore, social media analysis attracts more and more researchers every day [1]. Social media provide us with an enormous amount of data. This" Big Data" brings an opportunity for researchers to study various topics in this field. Sentiment analysis is one of the popular topics where the algorithms try to find people's opinions about individuals, issues, events, Etc. [2]. Profile gender identification [3], emotion analysis [4], popularity prediction [5], trend prediction [6], and bot detection [7] are some examples of trend topics in social media analysis.

One of the most important research topics in social media analysis is event detection approaches, which rely on clustering or classification algorithms with fixed temporal and spatial resolutions [8]. Most previous studies on event detection with social media focus on textual content. Harris et al. extracted users' geo-location and removed irrelevant tweets on Twitter [9]. Musaev et al. implemented a text classification algorithm to filter out noises and distinguish unrelated tweets from relevant ones on landslide detection [10]. Petersen et al. utilized hashtags to find tweets posted by health organizations during the Covid-19 pandemic [11]. Madani et al. applied a probabilistic model called Hierarchical Dirichlet Processes (HDP) [12], where for each tweet, the distribution of topics is calculated, and the one with the highest probability is considered a trending topic. Finally, tweets with similar trending topics are grouped into clusters. Also, Hou et al. integrated both textual and imagery content instead of employing the text-only approach and demonstrated that the precision of event detection is improved by 6.5% [13].

Various people are active on social media for different purposes. People's job and career affect their activities on social media. Thus, we expect it to predict someone's job using social media activities. As far as we know, there exists no study on predicting job title from any social media information or activity. Thus, this is the first attempt to predict the job or career of users from their social media activities. Job is a highly semantic target that may not be available in a specific part of a person's activities. For example, it may be available in bio, tweets, posts, profile pictures, Etc. Here, we focus on Twitter social media. Also, we assume that the job title has not been explicitly mentioned in any available information (for example, tweets). Job prediction using tweets could be considered a text classification problem [14]. The corresponding class of a text is usually not present in words but should be interpreted from the whole terms and their relations.

The core module for machine learning-based techniques in text classification is the embedding model, which attempts to convert words, phrases, or sentences to continuous-valued vectors. Bengio et al. proposed one of the first embedding models based on neural networks (NN) [15]. After that embedding models like word2vec [16], Elmo [17], GPT [18], [19], BERT [20], and Gshard [21] improved the embedding process. Then, the output of the embedding model is fed into a classifier for text classification. Different classifiers are employed by researchers, such as convolutional NNs (CNN) [22], recurrent NNs (RNN) [23], capsule NNs [24], and graph networks [25]. Studies such as tweet emotion recognition [26] or tweet gender identification [27] are also examples of text classification. Chiorrini et al. employed BERT as their embedding model and achieved 90% in emotion recognition accuracy. Baxevanakis et al. reached 70% accuracy in gender

978-1-6654-8087-1/22/$31.00 ©2022 IEEE

identification of twitterers employing TF-IDF and support vector machine (SVM) in the Greek language [28].

This paper proposed a method to predict jobs from Twitter profiles and tweets information. We used bios and hashtags as the input to our algorithms. As best we can tell, this is the first attempt to predict the job of twitterers. Thus, first, we needed to collect a dataset of users and their tweets. Next, we determined their job titles using Wikipedia crawling. Since each user may have several job titles, we proposed a clustering-based method to assign a unique job. We also presented two preprocessing algorithms to use emojis and hashtags in our classification method. Finally, we investigated the performance of three embedding models, i.e., TF-IDF, word2vec, and Glove [29] in combination with logistic regression, DNN, and CNN as classifiers. We achieved 54% of accuracy with a variety of TF-IDF and DNN algorithms. Our dataset and implementation are also available[1].

II. Dataset

A. Data Extraction

The first and most important part of this project is creating an appropriate dataset. Since we are working on a job prediction method based on analyzing the tweets, we need to pass through two vital steps:

1) Discovering celebrities: We need a list of celebrities with their Twitter accounts extracted by two searching algorithms. Details are discussed in the next section.
2) Job search: Each specific user needs a job title. So, we designed an algorithm to extract jobs by crawling the Wikipedia webpages. The users with no Wikipedia page would be removed from the list. More details are explained in the next session.

B. Data collection algorithm

Two approaches were designed for data gathering:

1) Search by emoji: we created a list of selected emojis utilized by the users on Twitter. Tweets and the corresponding usernames were the outputs of the searching algorithm. Each tweet has three primary attributes: the number of likes, retweets, and replies. We set a boundary of 50K on the minimum number of likes for the tweets to ensure the quality of our dataset.
2) Search by hashtags: We applied the same procedure in the previous section, replacing hashtags with emojis this time.

Our raw dataset consisted of the users' names and usernames by running the two above algorithms. Next, we extracted the corresponding bio and tweets for each username. Two limits were set for the tweets:

- All tweets must be written in English. So, we selected English-written tweets.
- Each user should have a minimum number of 10 and a maximum number of 40 tweets, so users have different numbers of tweets in the dataset.

Next, we proposed an algorithm to find job titles for each user. Since the job titles are our targets to predict, it is vital to have accurate labeling. That is why we focused on the celebrities' community because, in most cases, job titles could be found by analyzing the user's Wikipedia page. There is an infobox on Wikipedia, which summarizes that particular celebrity profile. Hopefully, the user's job titles are explained in the first two sentences of the Wikipedia summary. Let us take a look at one of them:" Katheryn Elizabeth Hudson, known professionally as Katy Perry, is an American singer, songwriter, and television judge. After singing in church during her childhood, she pursued a career in gospel music as a teenager." As is observed, the user's jobs are American singer, songwriter, and television judge. However, if we want to extract this information from our summaries, an algorithm must be designed. By analyzing these summaries, we realize that each phrase related to job titles is beginning with the verb " is" and ending with a dot (·), unless the user passed away or when searching for a group like music bands in which our phrase starts with the verb "are." We will explain the preprocessing steps to extract job information in the following.

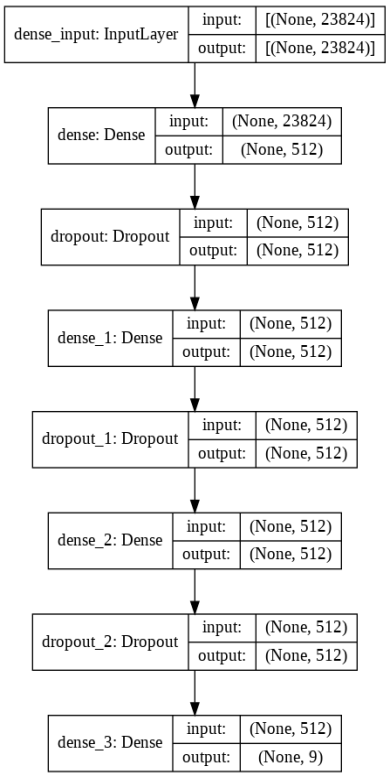

Fig. 1. DNN architecture consisted of 3 hidden layers, each having 512 nodes.

1) Removing nationalities: phrases like American-born or American-Italian-born were removed in this part.

[1] https://github.com/vassef/Job-Title-Prediction-from-Tweets-Using-WordEmbedding-and-Deep-Neural-Networks

2) Removing useless words: These are the words that will not provide us any useful information about the user's job titles. examples of these types of words are "professional" or "retired."
3) essential lists: These are adverbs, verbs, nouns, or structures that give more details about the specific job that the user is involved in." known for" expression is an "essential lists" in the "Lila Diane Sawyer is an American television broadcast journalist known for anchoring major programs" sentence. Our desire phrase is between the specific verb discussed earlier and the essential lists. So for this example, our desired phrase will be American television broadcast journalist.

In conclusion, we can extract each user's occupations, separating them by a comma.

III. PROPOSED METHOD

In this section, first, we tried to cluster job titles to determine a unique job for each user. The user information (tweets, bios, Etc.) was fed to the prepossessing steps. Finally, a classification model was trained to predict the job titles based on the available information. We used the K-means clustering algorithm for the first step so that the jobs with similar interests lay in the same group. To cluster the job titles, we needed an embedding model to compare the similarity between any two jobs. The Glove pre-trained model was utilized for this purpose. [29]. We used glove.6B.100d, where a vector with 100 elements shows each word. The dot product between word vectors equals the logarithm of the words' probability of co-occurrence.

Since each individual has several occupations, we may have several labels for a user. To overcome this, we selected the first two jobs of the users. In addition, each job may have several words, which should be converted to a single equivalent word. While some information would be lost here, it would help the generalization ability of our model. Suppose that the user's first two jobs are "video game programmer" and "designer." We can find the closest meanings as the distance between our desired word and the given words by employing the Euclidean distance to measure how far apart the two terms are. For instance, ranking the first five words by closeness to a given word, like "king," will result as [" king," "prince," "queen," "monarch," "brother"] which is pretty impressive. Now that we can rank the closest meanings to a single word, we can use the combination of math operations compatible with vectors, add or subtract several words together, and find the closet meanings for the aggregate of words. For instance, the "video game programmer" can be written as: "video+ game+ programmer" in the new vector representation, and the word "game" will be the closest equivalent word. So the user's new jobs will be "game" and "designer." Since we intended to cluster the job titles, and each sample of the clustering process is a 100-dimensional vector, the user's jobs consisting of two titles should have a constant length vector representation as to the ones with one job title. The applied method was Sum Word Vectors which adds the vectors of words in each document. [30]. Considering the previous example, the user's new jobs are "game" and "designer." By summing up the vectors assigned to the game and designer, we would have a unique vector representation of the user's jobs. So for each user, we will have a 100dimensional vector;

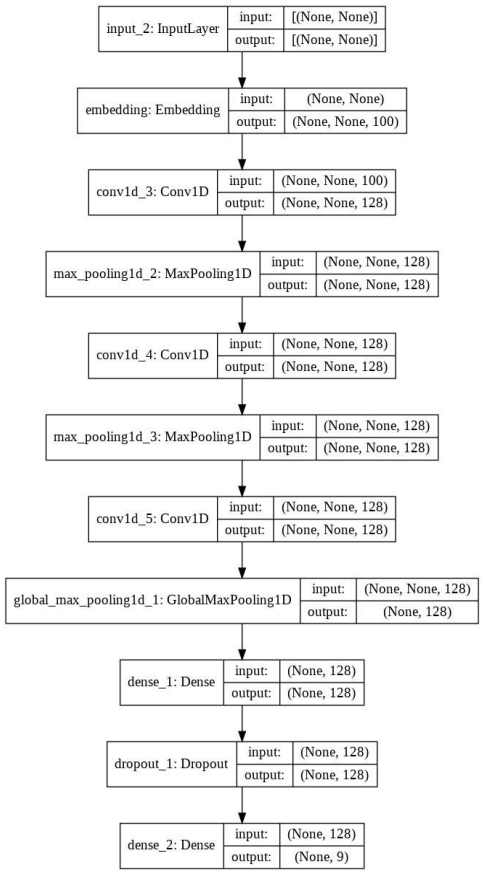

Fig. 2. CNN architecture consisted of three convolutional layers, each having a max-pooling at the end, finally passing through a feed-forward layer.

considering all the users, we will have a matrix with the dimension of 1314*100, passing through a clustering algorithm. We employed the elbow method [31] to find the optimal number of clusters. As Fig. 3 shows, the optimum number of groups is nine, meaning there are nine job titles. The most repetitive pattern inside each cluster was demonstrated in Table I. In this table, "mixed" means that various or unique job titles exist with low relation in that group. In the results section, we see how this condition affects the performance of the proposed method in labels 2 and 8.

There exist two items inside a text that can enhance our performance:

1) Emojis: Emojis inside a text are often used to show emotions, but some emojis can be related to jobs. There exist eight types of emojis:" Smileys People"," Animals-Nature"," Food-Drink"," Activity"," Travel-Places"," Objects"," Symbols", and " Flags". By some analysis, we found out that the most related category to jobs is the "Activity" type. Thus only all emojis of this type were

kept, and the rest were removed. Finally, each emoji was replaced by its meaning.

2) Hashtags: We have three approaches in facing hashtags:
   - Method 1(RHW): Remove both the hashtag sign (#) and the following word that leads to losing information.
   - Method 2(RH): Remove just the hashtag sign (#) that can insert some noises in our data since usually hashtags are written with no spaces, which causes ambiguity in the meaning.
   - Method 3(RHRW): Remove the hashtag sign (#) and replace the following word with its most relevant string.

We employed each of the above methods to compare the results in the next section. To find the most relevant string to each hashtag, we investigated all the samples (tweets or bios); the hashtag appeared at least once and considered the two most frequent words appearing in the user's bio as the most relevant string to that specific hashtag. Next, we substitute each hashtag with that string and reform our dataset. Finally, we do some cleanings by removing mentions, stopwords, punctuations, and URLs. Also, spell checking with lemmatization was done to prepare our dataset for the final step.

In the last step, we first created an embedding for each tweet. After that, embeddings were fed to our classification model. We investigated four popular embedding techniques named Bag of words, TF-IDF, Word2vec, and Glove with three classifiers: logistic regression, deep neural network (DNN), convolutional neural network (CNN). Logistic regression was implemented by multinomial class solved by the 'Newton-CG' algorithm. For DNN based classifier, we employed the TFIDF method as the embedding model. The network consists of 512 nodes in each hidden layer with considering drop out for generalization purposes as shown in Fig. 1. We employed the Glove pre-trained model for the embedding process in the third model. The output was fed to a CNN with three convolutional layers and a global max pooling at the end, finally passing through a fully connected part with 128 nodes, as shown in Fig. 2.

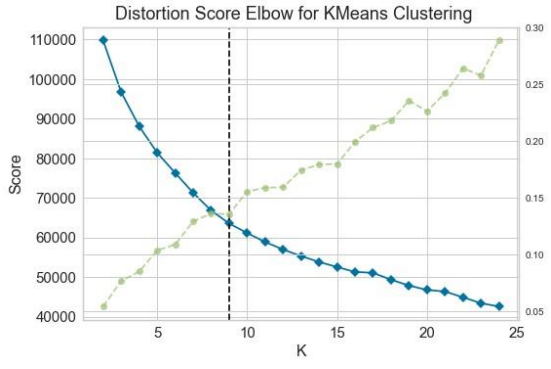

Fig. 3. The optimum number of clusters using the elbow method which happens at k=9.

IV. RESULT

As mentioned earlier, we tested three methods (combination of embedding models and classifiers). We also investigated the effect of hashtags on the results. As shown in Table II, deep learning methods malfunctioned on the small dataset (To be satisfied with more complicated models, we need to generate more data), mainly when we employed "RHW," in which we ignored all of the hashtags. Also, the model performance for each method would not differ when using simple models like logistic regression and DNN, but by making the model more complex (From DNN to CNN), the result would be worse. The second method ("RH") outperformed in comparison with "RHW" and "RHRW" when we used CNN as a classifier. However, "RH" and "RHRW" worked similarly in each case, though "RH" did better a little bit, meaning that increasing the influence of the user's bio by taking into account the relationship between hashtag and bio would not give us any superiority versus ignoring them. In conclusion, the bios and hashtags do not enhance the accuracy of the proposed model. Also, in Table III, we have a classification report of the DNN model with a 53% accuracy on the "RH" method consisting of three indicators: precision, recall, and the f1score.

I. TABLE I

PATTERNS FOR EACH LABEL

Label	*Pattern*
0	singe&songwriter
1	politician
2	singer&mixed
3	footballer
4	actor-actress&singer
5	basketball
6	actor-actress&comedian
7	rapper&singer
8	television&mixed

II. TABLE II

FINAL ACCURACIES FOR THE THREE METHODS

	Accuracy		
Model	*RHW*	*RH*	*RHRW*
Logistic	53.6	54	52.1
DNN	54	53	52
CNN	38	40	38

III. TABLE III

DNN REPORT FOR METHOD2(RH)

DNN classification report			
Label	*precision*	*recall*	*f1-score*
0	59	45	51
1	60	43	50
2	30	41	34
3	76	76	76
4	61	61	61
5	61	61	61
6	55	59	57
7	56	58	57
8	42	28	33
M-avg	56	52	53
W-avg	54	53	53

By analyzing the labels, the high performance of the third label with a precision of 76% was observed. The third label

shows the footballers group, meaning their tweets could easily discover this group. Labels 2 and 8 were below the average of the model performance, and the rest of the labels were on average. As shown in Table I, both were mixed groups where it was hard to assign a *shared label* or job title to them. We believe this diversity is why the performance is low in these two labels. Finally, we calculated the macro average (shown by m-average in Table III) and weighted average (shown by w-average in Table III) for each column in Table III.

V. CONCLUSION

In this paper, the possibility of job title prediction for Twitter users was investigated by using a dataset of users' tweets and bios. Then we added the user's job title using the Wikipedia information. A clustering method was used to label each user with the most relevant job title to overcome the challenge of multiple job titles per person. After that, various word embedding methods and classifiers were evaluated, and finally, a technique based on TF-IDF word embedding and a DNNbased classifier was proposed. Our results show that the job title of Twitter users could be predicted with 54% accuracy for nine unique job titles. Our accuracy is 43% better than the random classification, which for nine classes, the accuracy of a random classifier is $\frac{1}{9}$=11%.

ACKNOWLEDGMENT

The authors would like to offer their special thanks to Adak Vira Iranian Rahjo (Avir) company for providing needed hardware, including GPU, for this project.

IV. REFERENCES

[1] N. A. Ghani, S. Hamid, I. A. T. Hashem, and E. Ahmed, "Social media big data analytics: A survey," *Computers in Human Behavior*, vol. 101, pp. 417–428, 2019.

[2] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," *Knowledge and Information Systems*, vol. 60, no. 2, pp. 617–663, 2019.

[3] D. Kosmajac and V. Keselj, "Twitter user profiling: bot and gender identification," in *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 2020, pp. 141–153.

[4] F. M. Plaza-del Arco, M. T. Mart´ın-Valdivia, L. A. Urena-L˜ opez, and´ R. Mitkov, "Improved emotion recognition in spanish social media through incorporation of lexical knowledge," *Future Generation Computer Systems*, vol. 110, pp. 1000–1008, 2020.

[5] K. Wang, P. Wang, X. Chen, Q. Huang, Z. Mao, and Y. Zhang, "A feature generalization framework for social media popularity prediction," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 4570–4574.

[6] D. Rousidis, P. Koukaras, and C. Tjortjis, "Social media prediction: a literature review," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 6279–6311, 2020.

[7] J. Rodr´ıguez-Ruiz, J. I. Mata-Sanchez, R. Monroy, O. Loyola-Gonz´ alez,´ and A. Lopez-Cuevas, "A one-class classification approach for bot´ detection on twitter," *Computers & Security*, vol. 91, p. 101715, 2020.

[8] X. Dong, D. Mavroeidis, F. Calabrese, and P. Frossard, "Multiscale event detection in social media," *Data Mining and Knowledge Discovery*, vol. 29, no. 5, pp. 1374–1405, 2015.

[9] J. K. Harris, L. Hinyard, K. Beatty, J. B. Hawkins, E. O. Nsoesie, R. Mansour, and J. S. Brownstein, "Evaluating the implementation of a twitter-based foodborne illness reporting tool in the city of st. louis department of health," *International journal of environmental research and public health*, vol. 15, no. 5, p. 833, 2018.

[10] A. Musaev and Q. Hou, "Gathering high-quality information on landslides from Twitter by relevance ranking of users and tweets," in *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2016, pp. 276–284.

[11] K. Petersen and J. M. Gerken, "# covid-19: An exploratory investigation of hashtag usage on Twitter," *Health Policy*, vol. 125, no. 4, pp. 541–547, 2021.

[12] A. Madani, O. Boussaid, and D. E. Zegour, "Real-time trending topics detection and description from Twitter content," *Social Network Analysis and Mining*, vol. 5, no. 1, pp. 1–13, 2015.

[13] Q. Hou, M. Han, F. Qu, and J. S. He, "Understanding social media beyond text: a reliable practice on Twitter," *Computational Social Networks*, vol. 8, no. 1, pp. 1–20, 2021.

[14] J. Peng-tao and S. Wei, "A survey of text classification based on deep learning," *Computer and Modernization*, no. 07, p. 29, 2021.

[15] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The journal of machine learning research*, vol. 3, pp. 1137–1155, 2003.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[17] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[18] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[19] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020.

[22] B. Guo, C. Zhang, J. Liu, and X. Ma, "Improving text classification with weighted word embeddings via a multi-channel textcnn model," *Neurocomputing*, vol. 363, pp. 366–374, 2019.

[23] Y. Lan, Y. Hao, K. Xia, B. Qian, and C. Li, "Stacked residual recurrent neural networks with cross-layer attention for text classification," *IEEE Access*, vol. 8, pp. 70401–70410, 2020.

[24] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–221, 2020.

[25] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[26] N. Alswaidan and M. E. B. Menai, "A survey of state-of-the-art approaches for emotion recognition in text." *Knowledge & Information Systems*, vol. 62, no. 8, 2020.

[27] P. Vashisth and K. Meehan, "Gender classification using Twitter text data," in *2020 31st Irish Signals and Systems Conference (ISSC)*. IEEE, 2020, pp. 1–6.

[28] S. Baxevanakis, S. Gavras, D. Mouratidis, and K. L. Kermanidis, "A machine learning approach for gender identification of greek tweet authors," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2020, pp. 1–4.

[29] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[30] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[31] P. Bholowalia and A. Kumar, "Ebk-means: A clustering technique based on elbow method and k-means in wsn," *International Journal of Computer Applications*, vol. 105, no. 9, 2014.