iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51765

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

A Deep Learning Approach for Author Profiling using Word Embeddings

Dr. T. Raghunadha Reddy[1], B. Madhubala[2], G. Varshini[3], S. K. Fayaz[4]
[1]*Associate Professor, Department of CSE, Matrusri Engineering College, Hyderabad*
[2, 3, 4]*Student, Department of CSE, Matrusri Engineering College, Hyderabad*

***Abstract**: The task of author profiling involves predicting various characteristics of an author based on their writing style, such as their age, gender, native language, and personality traits. The PAN2013 shared task focused on author profiling in social media, where participants were tasked with predicting the gender and age of Twitter users based on their tweets. In recent years, deep learning approaches have become popular for author profiling. Two popular models are GloVe and FastText are used by the researchers to generate word embeddings. GloVe is a word embedding model that represents words as vectors in a high-dimensional space, while FastText takes into account subword information to represent words. Both models have been shown to be effective for various natural language processing tasks. For the PAN2013 task, participants used various deep learning models with GloVe and FastText embeddings to predict the age and gender of Twitter users. Some approaches used a combination of multiple models to improve the performance. In this article, we focused on improving the accuracy of age and gender classification on the PAN2013 dataset, which is a benchmark corpus for author profiling. We utilized deep learning models such as Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) classifiers to classify authors based on their age and gender. We also used pre-trained word embeddings such as FastText and GloVe to represent the text data. Our results showed that the LSTM model achieved an accuracy of 57.53% for age classification and 60.48% gender classification, while the CNN model achieved an accuracy of 59.32% for age classification and 52.21% for gender classification. We observed that these models have been shown to be effective for various natural language* **processing tasks and can be used for other author profiling tasks as well.**

Keywords: Author Profiling, Gender Prediction, Age Prediction, LSTM, CNN, Glove, FastText

I. INTRODUCTION

The author profiling (AP) task aims at identifying author demographics, such as age, gender, personality traits, or native language, basing on the analysis of text samples [1]. This research area has experienced an explosive increase in interest in recent years. It contributes to marketing, security, terrorism prevention, and forensic applications, among other. The approaches that tackle the task of AP from the machine-learning perspective view the task as a multi-class, single-label classification problem, when the set of class labels is known a priori. Thus, AP is modelled as a classification task, in which automatic methods have to assign class labels (e.g., male, female) to objects (texts) [2]. Machine-learning algorithms require input data to be represented in the form of a fixed-length feature vector. Approaches that have been used to obtain such vector include bag-of-words, bag-of-n-grams, etc., models.

From a deep learning perspective, author profiling involves training models to predict one or more predefined attributes of an author based on their writing style. This typically involves using neural network architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to learn complex patterns in the text data and extract features that are relevant for the prediction task [3]. For example, in a gender classification task, a deep learning model would be trained to predict the gender of an author based on their writing style. The model would be trained on a large dataset of text samples with known gender labels, and would learn to recognize patterns in the text data that are indicative of male or female writing styles. Similarly, in an age classification task, the model would be trained to predict the age of an author based on their writing style, and would learn to recognize patterns in the text data that are associated with different age groups. Deep learning models can be used to tackle a wide range of author profiling tasks, including but not limited to gender, age, language variety, personality traits, and sentiment analysis. By using complex neural network architectures, these models can learn to identify subtle patterns in the text data that might be missed by traditional machine learning approaches, and can achieve state-of-the-art performance on a wide range of benchmark datasets. In this article, we have used FastText and GloVe algorithms to obtain document embeddings for the AP task. We applied the LSTM and CNN classifiers to the embeddings, and conducted experiments on a single genre of data. To train our models, we split the dataset into training and validation sets, with a ratio of 80:20.

We then pre-processed the text data by removing stopwords and performing tokenization. Next, we used the pre-trained word embeddings to convert the text data into fixed-length vectors that could be used as input to our models. For gender classification, we trained a CNN classifier on the text data using the pre-trained FastText and GloVe embeddings. The CNN model is a type of neural network that can capture local features in the text data. We trained the CNN classifier using the binary cross-entropy loss function and the Adam optimizer. For age classification, we trained an LSTM classifier on the text data using the pre-trained FastText and GloVe embeddings. The LSTM model is a type of recurrent neural network that can capture long-term dependencies in the text data. We trained the LSTM classifier using the categorical cross-entropy loss function and the Adam optimizer.

FastText and GloVe are popular methods for generating word embeddings, which are representations of words as dense, low-dimensional vectors. However, there are some differences in their approach and specialties. FastText is an extension of the word2vec model and its key innovation is the ability to take into account subword information, such as prefixes and suffixes, in addition to whole words [4]. This allows FastText to generate embeddings for rare or misspelled words, as well as to capture the meaning of morphologically complex words. FastText also provides a way to generate embeddings for out-of-vocabulary words by combining their character n-grams. On the other hand, GloVe is a method for generating word embeddings that uses a co-occurrence matrix to capture the global context of words. GloVe stands for Global Vectors for Word Representation and its approach is based on the intuition that words that co-occur frequently in a large corpus of text are likely to have similar meanings [5]. GloVe embeddings are generated by factorizing a matrix of word co-occurrence probabilities, which produces embeddings that are good at capturing semantic relationships between words. This article is organized into 7 sections. The existing approaches proposed for author profiling are discussed in section 2. The dataset characteristics are presented in section 3. The word embedding methods and LSTM that are used in this work are explained in section 4. The section 5 explains the proposed method. The experimental results are discussed in section 6. The section 7 concludes this work with future enhancements.

II. LITERATURE SURVEY

The PAN evaluation campaign has been organized annually since 2013 to promote studies on author profiling and related tasks. The winning approaches of each edition have used a variety of features and feature representations, including lexical, stylistic, content-based, and deep learning-based features, and have achieved high performance on predicting the author's age, gender, native language, personality traits, and use of hateful language in various types of text data [6].

In the PAN 2014 edition, the task was to identify the author's age and gender, but with an expanded dataset including blog posts, tweets, and hotel reviews in both English and Spanish. The winning approach for the age prediction task on English and Spanish was based on a combination of lexical, stylistic, and content-based features [7]. For gender prediction on English and Spanish, the winning approach relied on a variety of features, including lexical, morphological, and syntactic features.

The PAN 2015 edition [8] introduced new author profiling tasks, including predicting the author's native language and personality traits. The dataset consisted of blog posts, tweets, and Facebook posts in multiple languages. For the native language prediction task, the winning approach used a combination of character n-grams and lexical features. For the personality trait prediction task, the best-performing approach was based on a combination of lexical and stylometric features.

In PAN 2016 [9], the task was again expanded to include prediction of the author's native language, gender, and personality traits, as well as the use of hateful language. The dataset included Twitter and Facebook posts in multiple languages. The winning approach for native language prediction used character n-grams and word embeddings, while the best-performing approach for gender and personality prediction relied on a combination of lexical, stylometric, and topic-based features. For the hateful language detection task, the winning approach was based on a deep learning model using character-level embeddings and bidirectional LSTMs.

The winning approaches have included ensemble-based classification, content-based and style-based features, and second order representations. In 2015, the task was extended to four languages, and in 2016, the focus shifted towards cross-genre age and gender identification. The best performing system used combinations of stylistic features and the second order representation. The use of distributed representations of words, such as word2vec embeddings, has been limited in AP research. The doc2vec algorithm, which learns neural network-based document embeddings, has shown promise in previous research. This paper evaluates different parameters of the doc2vec algorithm and compares its performance with traditional feature representations. The evaluation includes both single- and cross-genre AP settings.

III. DATASET CHARACTERISTICS

The corpus used for the task of determining age and gender was built using thousands of blog posts from a wide range of themes to provide a realistic representation of the diversity of topics people talk about. Since blog posts are used for search engine optimization and can be automatically generated by robots or be advertisements (chatbots), the corpus includes such posts [6].

Additionally, some texts from last year's PAN task on sexual predator identification were included to test the robustness of author profiling approaches and to unveil fake profiles. To enable multilingual settings, the corpus includes a Spanish part in addition to English, as both languages are widely used worldwide above. In this work, the experiment performed on English dataset. The English training dataset contains 236,600 dialogues of different types of authors. The corpus is balanced with 118,300 dialogues per gender. Table 1 shows characteristics of English training dataset.

TABLE I
THE CHARACTERISTICS OF DATASET

Type of Data	Male	Female
Total Number of dialogues	118300	118300
Dialogues of 10s Age Group Authors	8600	8600
Dialogues of 20s Age Group Authors	42900	42900
Dialogues of 30s Age Group Authors	66800	66800

IV. WORD EMBEDDINGS

The word embedding techniques are used for generating word vectors for words that are used in the dataset [9]. In this work, FastText and Glove methods are used for experimentation.

A. *Fasttext*

FastText is a library for efficient text classification and representation learning developed by Facebook's AI Research team [4]. It is based on the concept of word embeddings, which represent words as vectors in a high-dimensional space, and can be used to capture semantic and syntactic similarities between words.

FastText has several features, including:

1) Unsupervised learning of word embeddings: FastText can learn word embeddings from unannotated text data using the skip-gram or continuous bag-of-words (CBOW) models.
2) Text classification: FastText can be used to classify text into pre-defined categories, such as sentiment analysis or topic classification.
3) Language identification: FastText can identify the language of a given text.
4) Out-Of-Vocabulary (OOV) words: FastText can handle OOV words by using subword information to infer the meaning of unseen words.

FastText works by breaking words into subwords or n-grams, which are then used to construct word representations. For example, the word "cat" could be broken down into the subwords "cat", "ca", "at". The word representation for "cat" would then be the sum of the subword embeddings. FastText can be trained on large amounts of data, which allows it to capture the nuances of language use and improve the accuracy of text classification tasks.

B. *GloVe*

The GloVe (Global Vectors for Word Representation) model is another popular approach for learning word embeddings, similar to FastText [5]. It was introduced by Stanford researchers in 2014 and also uses a word-context matrix to learn representations for words. The key idea behind GloVe is to use a weighted least-squares regression to factorize the word-context matrix into a product of two lower-dimensional matrices, one representing the words and the other the contexts. These matrices are then used as the word embeddings. GloVe is trained on a large corpus of text and the resulting embeddings can be used for various NLP tasks, such as sentiment analysis, language translation, and question answering. In practice, the GloVe model is often pre-trained on large datasets like Wikipedia or Common Crawl and the resulting embeddings are then used as a starting point for downstream tasks.

The main difference between FastText and GloVe is that FastText uses subword information to learn embeddings for words, while GloVe only considers entire words. As a result, FastText may perform better on tasks where rare or unseen words are important, while GloVe may perform better on tasks where the focus is on more common words.

V. PROPOSED METHOD

The architecture of proposed method is displayed in Figure 1.

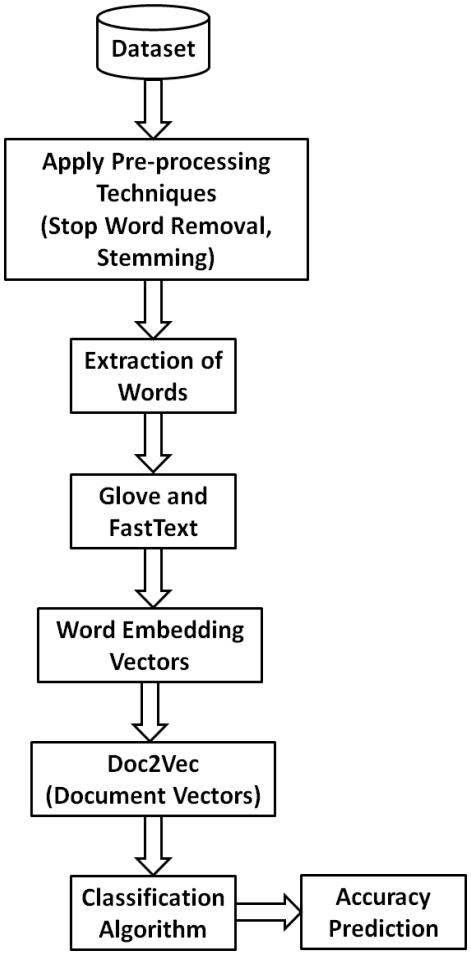


Fig. 1 The Proposed Method Architecture

In this work, first, collect the author profiling dataset from the PAN 2013 competition. Once the dataset is ready, apply different pre-processing techniques like punctuation marks removal, elimination of html tags , duplicate words, duplicate characters, commas, null values, lower case conversion, stop word removal and lemmatization. After cleaning the unwanted data from dataset, extract important words for experimentation. These words are passed to word embedding techniques like Glove and FastText for generating embedding vectors for words. The word vectors are used by the Doc2Vec method for generating the document vectors. The classification algorithms are trained with these document vectors to predict the performance of proposed method. In this work, the LSTM and CNN are used for classification.

A. LSTM

A Long Short-Term Memory (LSTM) model is a type of recurrent neural network (RNN) that is often used for natural language processing (NLP) tasks, such as sentiment analysis or language translation [3]. LSTM models are designed to address the vanishing gradient problem that can occur in traditional RNNs. The vanishing gradient problem occurs when the gradients used to update the model's parameters become too small to have a meaningful impact, which can result in the model being unable to effectively learn from the data. LSTMs address this problem by using a memory cell, which allows the model to selectively remember or forget information from previous timestamps.

In an LSTM model, the input data is first processed by an embedding layer, which converts each word in the input text to a vector representation.

The resulting vectors are then fed into one or more LSTM layers, which use the memory cell to selectively remember or forget information from previous timestamps. Finally, the output of the LSTM layers is passed through one or more dense layers, which produce the final output of the model. During training, the model's parameters are updated using Backpropagation Through Time (BPTT), which involves computing the gradients of the loss function with respect to the model's parameters at each timestamp and using those gradients to update the parameters. Overall, LSTM models have been shown to be effective for a wide range of NLP tasks, and are often used in industry and academia for tasks such as sentiment analysis, language translation, and speech recognition.

Working of LSTM model

1) The input text is pre-processed, including tokenization and possibly other steps such as stemming or stop word removal.
2) The reprocessed text is then fed into the LSTM model, which processes the text one time step at a time.
3) During each time step, the LSTM model updates its internal state based on the input at that time step and its internal memory of previous time steps.
4) Once the model has processed the entire input text, the final output of the model is produced, which can be used for tasks such as sentiment analysis or language translation.
5) During training, the model's parameters are updated using backpropagation through time (BPTT), which involves computing the gradients of the loss function with respect to the model's parameters at each timestep and using those gradients to update the parameters.

The trained model can then be used to make predictions on new input text. This LSTM model is a type of recurrent neural network (RNN) that is designed to analyse sequential data such as text data. The architecture of the model consists of several layers:

a) Input layer: This layer accepts two inputs - one for the text data and another for the numeric features.
b) Embedding layer: This layer converts the input text data into a dense vector representation using pre-trained word embeddings. It maps each word in the text to a vector of real numbers.
c) LSTM layer 1: This layer processes the word vectors sequentially and returns a sequence of hidden state vectors. It has 32 memory cells and returns the output at each time step.
d) Dropout layer: This layer applies dropout regularization to the output of LSTM layer 1 to prevent overfitting.
e) LSTM layer 2: This layer takes the output of the dropout layer and returns a single output vector. It has 16 memory cells and does not return the output at each time step.
f) Dense layer 1: This layer accepts the numeric input features and applies a linear transformation followed by the ReLU activation function. It has 32 units and applies L2 regularization to prevent overfitting.
g) Dense layer 2: This layer applies another linear transformation followed by the ReLU activation function. It has 16 units and also applies L2 regularization.
h) Concatenate layer: This layer concatenates the output of LSTM layer 2 and dense layer 2 to combine the information from both inputs.
i) Output layer: This layer applies a sigmoid activation function to the concatenated output to predict the gender of the user.

The model is trained using binary cross-entropy loss and is optimized using the Adam optimizer. The metrics used to evaluate the performance of the model are accuracy, precision, recall, and F1-score. Additionally, the model uses two callbacks - EarlyStopping and ReduceLROnPlateau - to prevent overfitting and improve convergence during training.

B. CNN

CNN algorithm is based on various modules that are structured in a specific workflow that are listed as Input, Convolution Layer (Kernel), Pooling Layer, Classification—Fully Connected Layer and Architectures [11]. Overview of the architecture for text classification using CNN layers are

1) *Input Layer:* The first layer of the model is the input layer which receives the text data as input.
2) *Convolutional Layer:* The convolutional layer performs convolution operation on the input data. The main goal of the convolution layer is to extract important features from the text data. The layer consists of several kernels, each of which extracts a different feature from the input.
3) *Pooling Layer:* The pooling layer reduces the dimensionality of the output from the convolutional layer. It helps in extracting the most important features from the input. There are different types of pooling layers such as Max Pooling, Average Pooling, etc.

4) *Fully Connected Layer:* The output from the pooling layer is flattened and then fed into the fully connected layer. The fully connected layer is a neural network layer that is used to classify the text data into different classes. The layer consists of neurons that perform a weighted sum of the inputs and apply a non-linear activation function to the result.
5) *Output Layer:* The output layer of the model is responsible for generating the final output of the model. It consists of neurons that produce the final predictions based on the input data.

VI. EXPERIMENTAL RESULTS

In this work, the experiment carried out for predicting the author's demographic features like age and gender based on the analysis of their written text. The LSTM and CNN models are used for classification and Glove and FastText are used for generating word embedding vectors. The accuracies of proposed for age and gender prediction are displayed in Table 2.

TABLE III
THE ACCURACIES OF PROPOSED METHOD FOR GENDER AND AGE PREDICTION

	CNN		**LSTM**	
Profile / Word Embedding Techniques	Glove	FastText	Glove	FastText
Gender	50.84	52.21	57.72	60.48
Age	57.68	59.32	55.39	57.53

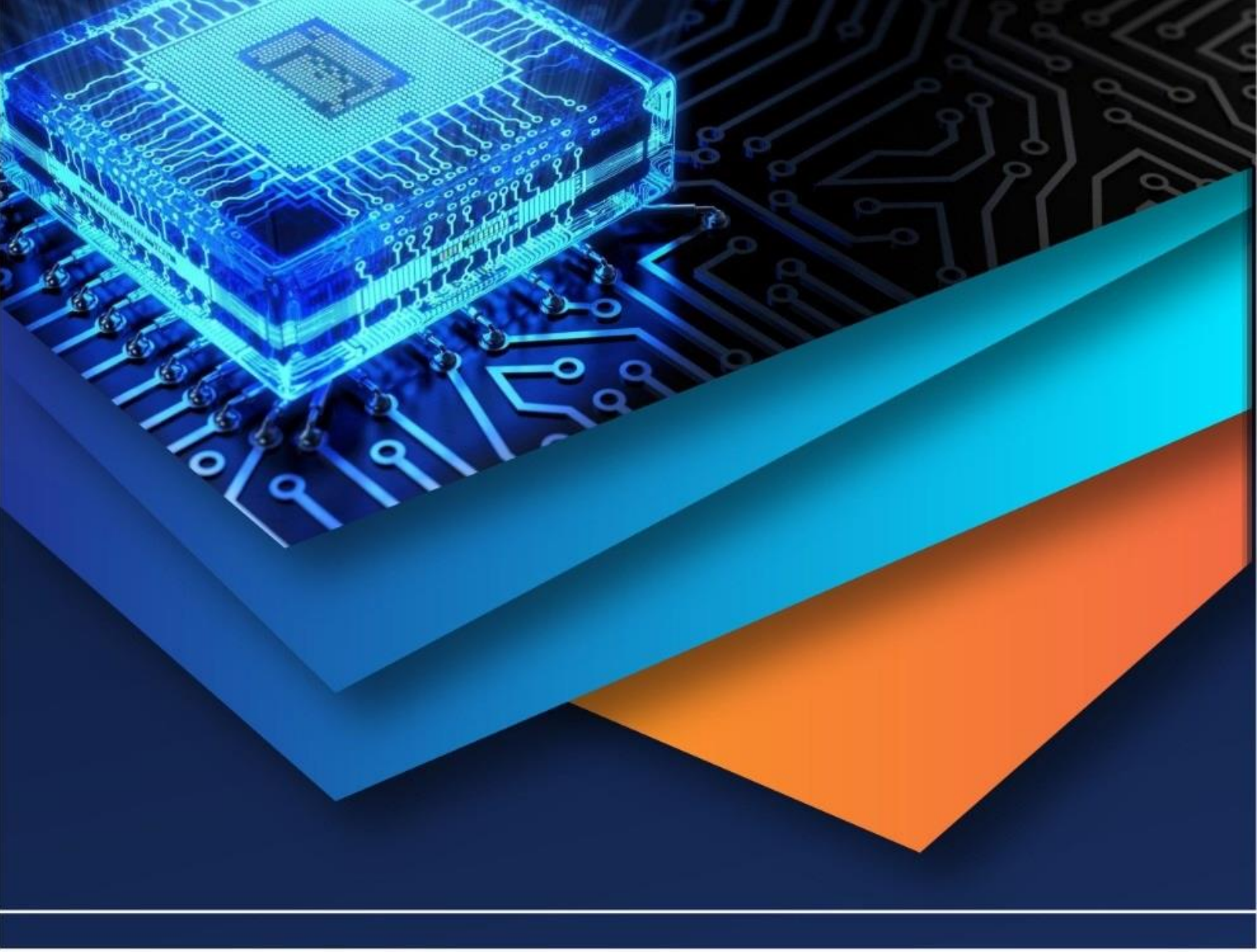
In Table 2, the LSTM model shows good performance than the performance of CNN model for gender prediction, but for age prediction, the CNN model shows good performance than LSTM model. The FastText embedding technique shows good performance than the performance of Glove embedding technique. The combination of LSTM and FastText model attained best accuracy of 60.48% for gender prediction. The combination of CNN and FastText model attained best accuracy of 59.32% for age prediction.

VII. CONCLUSION AND FUTURE SCOPE

Author profiling is a method of predicting the demographic characteristics like gender, age, nativity language, location, education background of authors by analysing their written text. In this work, we proposed a method by using word embedding techniques of Glove and FastText for generating word embeddings and LSTM and CNN are used as classification methods. The combination of LSTM and FastText model attained best accuracy of 60.48% for gender prediction. The combination of CNN and FastText model attained best accuracy of 59.32% for age prediction. In future work, we are planning to implement other word embedding technique of BERT for generating word embedding vectors and GRU and BERT as classification models. We are also planning to predict other demographic characteristics like location and nativity language of authors.

REFERENCES

[1] Raghunadha Reddy T, Vishnu Vardhan B, Vijayapal Reddy P, "A Survey on Author Profiling Techniques", International Journal of Applied Engineering Research, March 2016, Volume-11, Issue-5, pp. 3092-3102.
[2] Raghunadha Reddy T, Vishnu Vardhan B, Vijayapal Reddy P, "Profile specific Document Weighted approach using a New Term Weighting Measure for Author Profiling ", International Journal of Intelligent Engineering and Systems, 9 (4), pp. 136-146, Nov 2016.
[3] Roy Khristopher Bayot, Teresa Gon_calves, Multilingual Author Profiling using LSTMs Notebook for PAN at CLEF 2018
[4] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2017). Enriching word vectors with subword information, Transactions of the association for computational linguistics 5: 135–146.
[5] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," en, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162. [Online]. Available: http://aclweb.org/anthology/D14-1162 (visited on 01/28/2021)
[6] Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: Overview of the author profiling task at PAN 2013. In CLEF Conference on Multilingual and Multimodal Information Access Evaluation, CELCT, pp. 352-365 (2013).
[7] Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B.,Daelemans, W.: Overview of the 2nd author profiling task at pan 2014. In CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014, pp. 1-30 (2014).
[8] Rangel, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd Author Profiling Task at PAN 2015. In CLEF p. 2015 (2015).
[9] F. Rangel, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast, and B. Stein, "Overview of the 4th author profiling task at PAN 2016: Cross-genre evaluations," CEUR Workshop Proc., vol. 1609, pp. 750–784, 2016.
[10] Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.
[11] Kim, Y: Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY