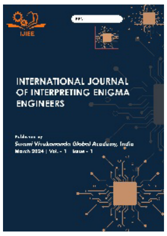
SENTIMENT ANALYSIS OF CODE-MIXED LANGUAGES


INTERNATIONAL JOURNAL OF INTERPRETING ENIGMA ENGINEERS

Padaga Sri Sai Sankar*,Dr.J.Avanija

Original Article

Sree Vidyanikethan Engineering College,Tirupati, 517102, India

*Corresponding Author's Email: saicorporate26@gmail.com and avans75@yahoo.co.in

Abstract

In the multilingual context of today, code-mixed language is frequently utilized. It happens when a sentence contains both foreign language vocabulary and grammar. Finding the sentence's polarity value is the aim of sentiment analysis of code-mixed language. It is mostly concerned with sentiment analysis of tweets that contain extra Hindi and English words and symbols. The collection is composed of 20,000 tweets, which produces word-level representations of the tweets for use as input in several models, including CNN, LSTM, and Bi-LSTM. When compared to other models, the Bi-LSTM model performs better. The precision of CNN, LSTM and Bi-LSTM is 65.00%, 58.59% and 54.24% respectively.

Keywords: Convolutional Neural Networks:, Long-Short term memory:, Sentiment Analysis:, Activation function.

Introduction

Natural language processing has an area called sentiment analysis. Numerous activities can be accomplished with it, such as text analysis, opinion mining, user modeling, text tone analysis, and online trend curation. Many names, including sentiment analysis, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, and review mining, are used in the literature, and so forth, to describe relatively distinct tasks. All of these are categorized as opinion mining or sentiment analysis, though. Depending on the sort of mixing, the switching between the languages is referred to as either code-switching or code-mixing. Compared to sentence-level code mixing, word-level code mixing occurs more frequently. India is a bilingual country where people who speak more than one language converse with each other in different languages. These people are not native English speakers. Depending on the sort of mixing, the transitioning between the languages is referred to as either code-switching or code-mixing. Word-level code mixing is more-greater than code mixing at the sentence level. Hinglish, an amalgam of Hindi and English, is more common in the northern part of India.

As an illustration, "Waglenikhil U perceived religion and caste in them. The country trusted them because they had talent Tum paida hi ulte hue the is the problem! All of the words in the sample above are written in Roman script, even if some are written in Hindi and some in English. Social media texts have created a plethora of new opportunities and difficulties for language technology and information access, including blogs, microblogs, and conversations (like Facebook Messenger and WhatsApp). As a result, they are currently one of the main areas of research. Even though the majority of language technologies in use today are designed with English in mind, non-native English speakers still mix English with other languages when using social media.

Code-mixing poses a number of hidden challenges to natural language processing (NLP) activities, including part-of-speech tagging, machine translation, dependency parsing, word-level language identification, and semantic processing. Sentiment analysis gets more challenging when social media data is obtained and contains noise. Text that has been code-mixed incorporates vocabulary and grammar from other languages. Sentiment analysis faces a hurdle because conventional semantic analysis methods are unable to fully convey the meaning of the sentences. The phrases' brief, condensed info presents another difficulty. The fact that the same words may appear in the sentence in multiple ways is another restriction. To resolve these problems, preprocessing tasks need to be finished. This work's main goal is to preprocess tweets and classify them as neutral, bad, or positive.

Literature Review

The most widely used text categorization tool, according to the author, is sentiment analysis, which analyzes incoming messages to ascertain whether their underlying sentiment is positive, negative, or neutral. When applied imaginatively, sophisticated AI algorithms can be an effective tool for carrying out in-depth study. The author used Intelligent Systems and Applications in the proposed system to recognize the [1] Sentiment Analysis.

Instead of learning character or word level representations, the author proposed that the LSTM (Subword-LSTM) architecture allows learning sub-word level representations. In our design, we make use of a linguistic prior to help us estimate the sentiment value of significant morphemes. Our model's learning of morpheme-level feature maps indicates that this also appears to work effectively in highly noisy text that contains misspellings. Furthermore, conjecture that the better performance is caused by using the Subword-LSTM architecture [2] to encode this linguistic antecedent. For sentiment analysis in Hi-En code-mixed text, it outperforms the present system by 18% and delivers 4-5% higher accuracy than conventional methods on our dataset.

The author proposed Twitter dataset sentiment categorization. For sentiment analysis, we employ several deep learning and machine learning techniques. Using five of our best models, we finally employ a majority vote ensemble approach to obtain 83.58% classification accuracy on the Kaggle public [3] leaderboard. To attempt to classify the polarity of a tweet as positive or negative, try running sentiment analysis on "tweets" using a range of machine learning methods. The tweet's predominant sentiment should be chosen as the final label if it contains both positive and negative aspects.

The author proposed Sentiment analysis is crucial for numerous real-world applications, including recommendation systems, stance identification, review analysis, and more. Sentiment analysis gets more challenging when social media data is used to collect noisy data. India is a multilingual nation where individuals converse among themselves in multiple languages. Depending on the sort of mixing, the process of switching between the languages is referred to as either code-switching or code-mixing. An overview is given of the cooperative challenge on sentiment analysis of code-mixed Hindi-English data pairs gathered from various social media channels. The task, dataset, baseline, evaluation, and [4] participant's systems are all described.

The author proposed a Sentiment analysis is a well-known discipline in which sentiment, opinion, and emotion are recognized and categorized in written text by humans. Previous polarity lexicons are the starting point of a common computational approach to sentiment analysis, where objects are marked with the prior out-of-context polarity that individuals understand through the application of their cognitive skills. The majority of research projects in the sentiment lexicon literature focus on texts written in English. suggested a variety of computational methods for creating SentiWordNet(s) for Indian languages, including [5] techniques based on WordNet, dictionaries, corpuses, and generative models. SentiWordNet(s) for the Indian languages Telugu and Hindi are currently being developed.

The author proposed a system where the Python library scikit-learn integrates a range of state-of-the-art machine learning methods for medium-scale supervised and unsupervised applications. This package attempts to familiarize non-specialists with machine learning through the use of a high-level, general-purpose language. [6] The focus is on documentation, consistency of APIs, performance, and ease of use. Its distribution under the streamlined BSD license and minimum dependencies promote its use in both commercial and academic contexts.

The author proposed a system in which In many sentiment analysis applications, Sentiment Word Identification (SWI) is a fundamental technique. The majority of current studies use seed words, which results in poor robustness. presented a brand-new SWI model that is based on optimization. Our methodology, in contrast to earlier methods, uses the sentiment labels of documents rather than seed words. Experiments conducted on [7] real datasets demonstrate that in terms of efficiency, WEED works better than the most advanced seed word techniques.

Sentiment Analysis of Code-Mixed Text (SACMT) is a unique approach that classifies sentences into positive, negative, or neutral sentiments using contrastive learning which was proposed by the author. Map phrases from standard and code-mixed languages to a single sentiment space using the shared parameters of siamese networks. Additionally, offer a fundamental preprocessing technique based on clustering to find variants of transliterated words with mixed codes. Experiments reveal that SACMT performs 7.6% more accurately and 10.1% better on F-score than the most complex algorithms [8].

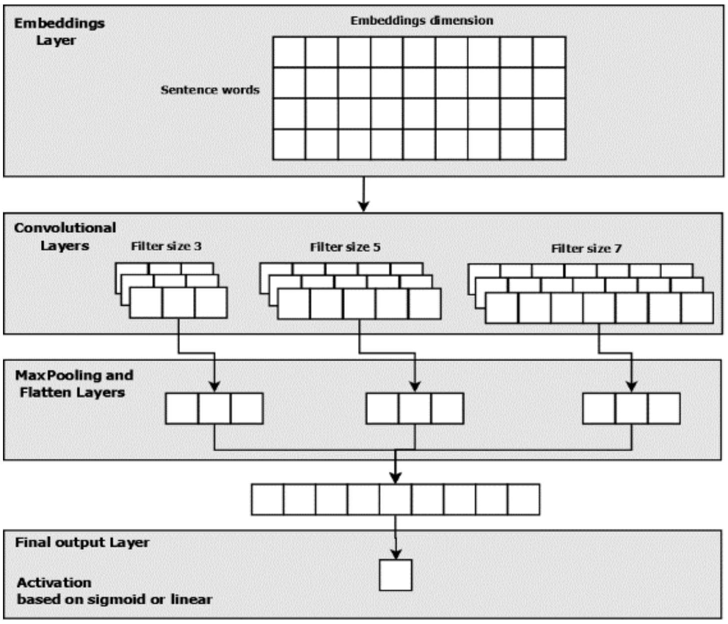
Proposed Approach

Sentiment Analysis of Code-Mixed Text (SACMT), a unique methodology presented by Nurendra Choudhary in 2018, uses contrastive learning to classify sentences into positive, negative, and neutral attitudes. The mutual characteristics pertaining to Siamese networks describe the standard and code-mixed language sentences in a typical sentiment space. A hybrid architecture is developed for sentiment analysis of mixed data using Hindi and English codes. In 2019, Ivan Provilkov presented the BPE-dropout technique. The BPE dropout technique is a straightforward and efficient sub-word regularization method that works with traditional BPE. This approach performs better on a variety of translation tasks than both BPE and earlier subword regularization.

PROPOSED SYSTEM

Convolutional neural networks (CNNs), long short-term memory (LSTMs), and bidirectional LSMs have all been implemented (Bi-LSTM). In convolutional neural network models, the outputs are computed by applying convolutions over the inputs. In CNNs, various filters are applied to the data by each layer, and based on the task at hand, the model determines the values of these filters. This procedure culminates in the combination of each layer's output. Figure 1 displays the [9] CNN model's design. Since the unidirectional LSTM model has only ever seen inputs from the past, it maintains historical information. Connecting hidden layers pointing in opposite directions to the same output is known as the bidirectional LSTM model. Data from both past and future states is accessible to the output layer. Figure 1 depicts the architecture of the CNN.

Figure 1: Proposed CNN architecture



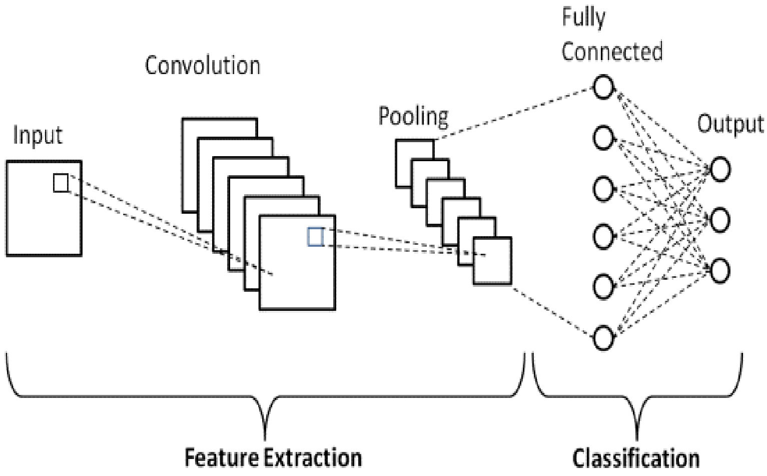
Convolutional Neural Network (CNN)

The primary applications of convolutional neural networks, which are deep learning neural networks, include image processing and classification. Because information only flows in one direction from CNNs' inputs to their outputs, they are feedforward networks. Comparatively speaking to other classification algorithms, this one requires less pre-processing to distinguish one image from another. CNN's main benefit over other systems is its automatic feature detection, which eliminates the need for human supervision. CNN architecture uses three different types of layers: fully connected, pooling, and convolutional. A fully connected layer comes last, followed by a possible number of convolutional and pooling layers after the [10] first convolutional layer. A deep model is created by stacking these layers on top of one another. Figure 2 shows the general architecture of CNN.

- **Convolutional layer**: Layer of convolution Layer serves as a feature extractor, taking features out of the input image. Its trainable weights, or kernels, are a matrix of integers that serve as learnable filters. To construct a feature map, the filter moves across the image in strides and performs a dot product with the area it is hovering over. Multiple features are [11] extracted at each position due to the varying weights assigned to different feature maps inside the same convolutional layer.
- **Pooling layer:** A pooling layer reduces the dimensionality of the feature maps by selecting the best features. Unlike the convolutional layer, the pooling operation sweeps across the entire input without using weights within the layer of pooling. The filter creates an output array by running the values in the corresponding fields through an aggregate function. Two primary categories of pooling exist:

  i) Max Pooling: This method uses a function to select the pixel with the greatest value within the input area.
  ii) Average Pooling: This feature includes a function that averages all of the pixels in the select input section and outputs the result to the output array. Max Pooling is utilized in the suggested approach to reduce the dimensionality of the image.

- **Fully Connected layer:** Every neuron in one layer is connected to every other layer by a basic feed forward neural network. This layer receives the output from the pooling layers that has been flattened. This layer helps to classify the images by acting as a classifier. For picture categorization, it makes use of the Sigmoid and SoftMax activation [12] functions. The suggested technique, which is popular for classifying photos, uses the SoftMax activation function to classify the images by generating probability distributions. Figure 2 shows the layered CNN architecture.



The dataset comprises 20,000 code-mixed tweets and associated emotions. It is a part of a task on the open-source web platform Codalab's SentiMix Hindi-English Competition. There are 17,000 [13] tweets in the training dataset and 3000 in the test dataset. Every tweet in the training dataset begins with "meta," and each one is associated with a distinct id, polarity, and sentiment value (positive, negative, or neutral) that represents the [14] sentiment. Each word in a tweet has its corresponding language designated; for instance, Hin stands for Hindi, Eng for English, and O for other symbols. The training, validation, and testing tweets are included in Table 1.

Table 1: Tweet Samples used for model building

Language Used	Tweet Samples	Validation Samples	Test Samples
Hin-Eng	1400	3000	300

Steps for Proposed System:

Step-1: Loading the Dataset

Step-2: Pre-Processing the Dataset

- Converting tweet into lowercase
- Replace contractions
- Demojize emojis
- Swap out characters that repeat with a maximum of two characters.
- Tokenization

Step3: Training the model

CNN Model:

- Embedding Dimensionality
- Parameters: Input Dimensions, Output Dimensions, Input Length.
- Conv1D
- Parameters: Filters, Kernal Size, Activation Function (ReLu)
- y = max (0, x)

GlobalAveragePooling1D:

Given, the 1D Global Average Pooling block calculates the maximum of all the (input size) values for each of the (input channels), [15] resulting in a 2-dimensional tensor tensor of size (input size) x (input channels). In order to receive a one-dimensional input for the 1D Global max pooling block, utilize a reshape block with a Target shape of (input size, 1).

Dense Layer (Hidden Layer):

Units, Activation Function (ReLu)

Dense Layer (Output Layer):

Units, Activation Function (Softmax)

Step4: Predicting with model

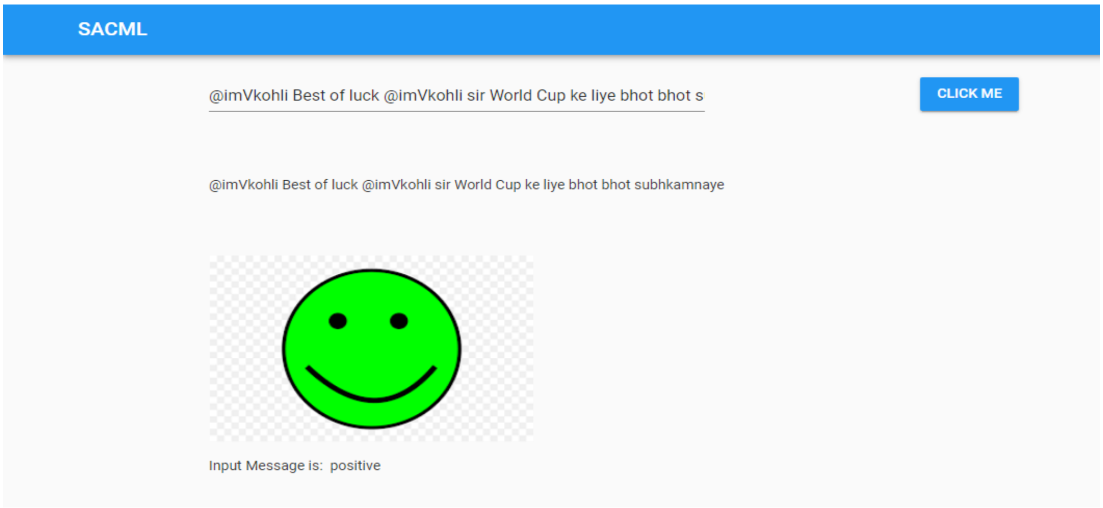
Step5: Deployment of the model

The model output will say that the comment is positive or negative or neutral

[Supporting comments --> Positive, Highly negative or bad comments --> Negative, Not supporting or not disagreeing --> Neutral]

Figure 3 specifies the Sample Input1.

Figure 3 Input and Output Image



@imVkohli Best of luck @imVkohli sir World Cup ke liye bhot bhot s

@imVkohli Best of luck @imVkohli sir World Cup ke liye bhot bhot subhkamnaye

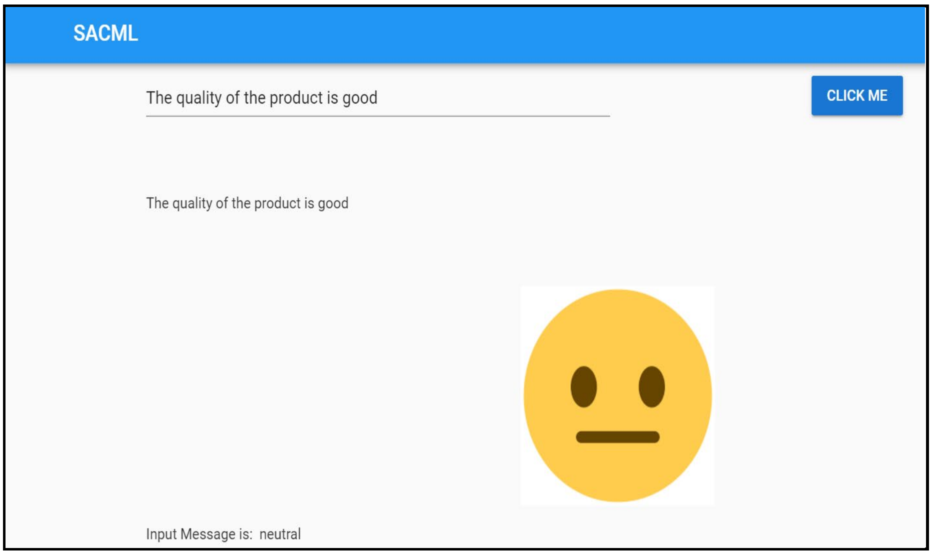
Input Message is: positive

Output: [Positive]

Comment Convey: Supporting comment

Figure 4 specifies the Sample Input2:

Figure 4 Input and Output Image



The quality of the product is good

The quality of the product is good

Input Message is: neutral

Output: [Neutral]

Comment Convey: Not supporting or agreeing

Figure 6 Graph of CNN model accuracy

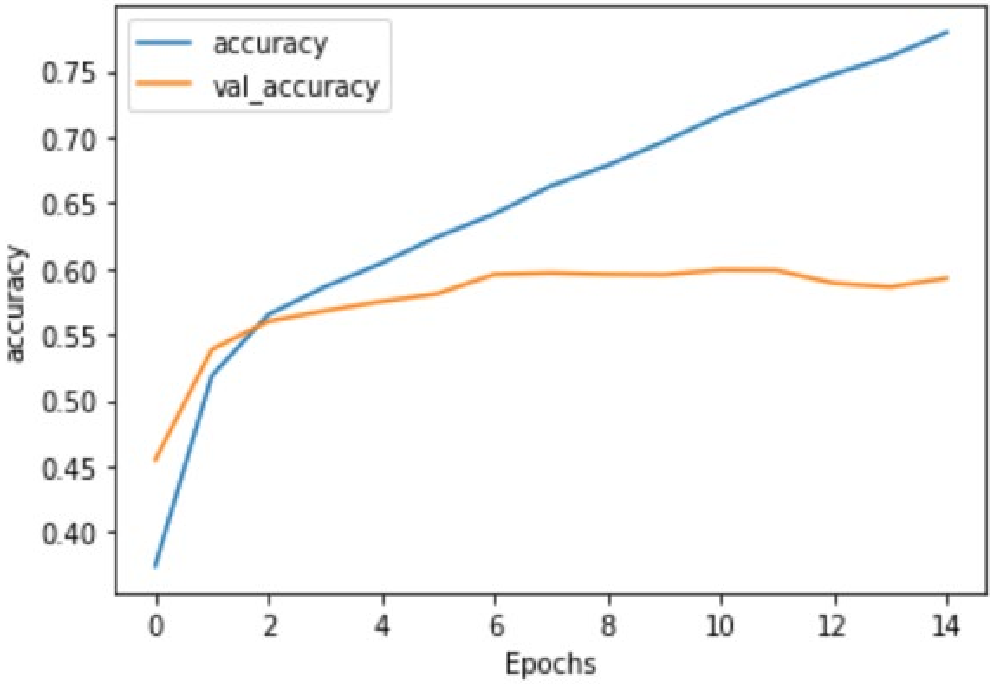


Figure 6 specifies the accuracy of the CNN model based on number of epochs.

Conclusion And Future Work

The suggested technique predicts the feelings of code-mixed languages using several neural network [16] models. The Tweet dataset is used by the model. With a small dataset, the CNN model performs exceptionally well, with an accuracy of 65%. The LSTM and Bi-LSTM models also perform well, with respective [17] accuracy rates of 58.59% and 54.24% for the provided dataset. Our approach primarily focuses on sentiment analysis of tweets that contain additional symbols and words from the Hindi and English languages.

Based on the provided experimental review, several other studies can be conducted in the future. To increase the models' accuracy, one option is to train them using a huge dataset. This activity can be expanded to [18] include writing comments on various social media platforms, such as Facebook, Instagram, WhatsApp, and so on.

References

1. Aditya Joshi, P. A. (2016). Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. ArXiv , arXiv:1611.00472.
2. Abdul Fatir Ansari, A. S. (2017). Twitter Sentiment Analysis.
3. Abney, B. K. (2014). Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods. Proceedings of the 2013 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics. Birdsong, T. (2018, jan 13). 2018 Texting Slang Update: How to Decode What Your Teen is Saying Online. Retrieved from mcafee.com:https://www.mcafee.com/blogs/consumer/family-safety/2018-texting-slang-update-decodeteen-saying-online/
4. Braja Gopal Patra, D. D. (n.d.). Sentiment Analysis of Code-Mixed Indian Languages: An Overview of SAIL_Code-Mixed Shared Task @ICON-2017.

5. Bandyopadhyay, A. D. (2010). SentiWordNet for Indian Languages. Das2010SentiWordNetFI.
6. Fabian Pedregosa, G. V. (2012). Scikit-learn: Machine Learning in Python. CoRR, abs/1201.0490.

7. Hongliang Yu, Z.-H. D. (2013, 08). Identifying Sentiment Words Using an Optimizationbased Model without Seed Words. ACL 2013, 2, 855-859.
8. Ivan Provilkov, D. E. (2019). BPE-Dropout: Simple and Effective Subword Regularization. ArXiv.
9. Klenner, M., Tron, S., Amsler, M., & Hollenstein, N. (2014). The Detection and Analy- sis of Bi polar Phrases and Polarity Con icts. Proceedings of 11th International Workshop on Natural Language Processing and Cognitive Science, Venice, Italy, 2014 - 2014. ZORA: Zurich Open Repository and Archive, University of Zurich ZORA.
10. Kim, E. (2006). Reasons and Motivations for Code-Mixing and Code-Switching. 4 (EFL).
11. Liu, B. (2012). Sentiment Analysis and Opinion Mining. calofornia: Morgan & Claypool Publishers, May 2012.
12. Monkeylearn.com. (n.d.). Retrieved from Monkeylearn.com: https://monkeylearn.com/sentiment-analysis-examples/
13. Nurendra Choudhary, R. S. (2018, april 3). Sentiment Analysis of Code-Mixed Languages leveraging Resource Rich Languages.
14. Nurfadhlina Mohd Sharef, H. M. (2016). Overview and Future Opportunities of Sentiment Analysis Approaches for Big Data. Journal of Computer Science, 12, 153-168.
15. P. V. Veena, M. A. (2018). Character Embedding for Language Identification in HindiEnglish Code-mixed Social Media Text.
16. Pruthwik Mishra, P. D. (2018). Code-Mixed Sentiment Analysis Using Machine Learning and Neural Network Approaches. ArXiv , abs/1808.03299}.
17. Piotr Bojanowski, E. G. (2017). Enriching Word Vectors with Subword Information. ArXiv.International Journal of Advanced Science and Technology Vol. 30, No. 1, (2021), pp. 01-11 ISSN: 2005-4238 IJAST 11 Copyright Ⓒ 2021 SERSC
18. Preslav Nakov, A. R. (2016). Proceedings of the 10th International Workshop on Semantic Evaluation. SemEval-2016 Task 4: Sentiment Analysis in Twitter. San Diego, California}: Association for Computational Linguistics.