JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Sentiment Recognition of Hinglish Code Mixed Data using Deep Learning Models based Approach

Shubham Das
M.Tech. (Data Science)ASET, Amity University
Noida, India shubham.das@s.amity.edu

Dr. Tanya Singh
Professor
*ASET, Amity University*Noida, India tsingh2@amity.edu

***Abstract*— The popularity of sentiment analysis in social media content has grown over the past several years as a result of its many applications in research on human-computer interaction, consumer behaviour, psychology, smart systems, etc. This problem has received a lot of focus due to the enormous amounts of data that are accessible via social media, which is frequently utilised to express thoughts and ideas. This study will use a labelled Hinglish dataset to pinpoint emotions. By using transformer-based models and multilingual word embeddings produced from FastText approaches, deep learning-based techniques are used to recognise emotions in tweets with mixed Hindi-English coding. Many deep learning models have been used to assess attitudes, including convolutional neural networks (CNN), long short-term memory (LSTM), bi-directional long short term memory (Bi-LSTM), and other transformer models like BERT. Convolutional neural networks (CNN) outperformed other models in terms of accuracy, coming in at 75.25%.**

Keywords*—NLP, FastText, convolutional neural networks (CNN), long short term memory (LSTM), and bi-directional long short term memory (Bi-LSTM), Bidirectional Encoder Representations from Transformers BERT.

I INTRODUCTION

People today connect online more than ever because to the popularity of social media. As a result, enormous amounts of textual data are produced, creating fascinating NLP challenges. The automatic recognition of a wide range of vocal expressions, including irony, anger, sarcasm, aggression, etc., is currently the subject of substantial research. Another topic that has drawn the attention of NLP experts is how to extract a person's emotions from the texts. To improve human-computer interaction, it is now more crucial than ever to recognise emotions in texts [1]. Several techniques, including text-based approaches, facial expression detection, and voice, can be used to identify emotions [2][3].

The science underlying the notion and research of text-based emotion identification is founded on the tenet that uplifting language is used by happy people. When expressing unfavourable emotions like rage, annoyance, or upset, a specific phrase having a negative connotation is employed. Contrary to what is commonly believed, emotions are an important part of human creativity and decision-making. In the era of artificial intelligence and increased interest in human-machine contact, a smart machine will need to be able to completely comprehend human emotions in order to communicate with humans successfully. Interest in emotional computing has risen recently as the focus has shifted to emotion recognition [4].

Due to how simple it is to obtain a sizable corpus of labelled data, the majority of prior research has been done on a monolingual dataset [5][6]. However, social media communication on many languages is quite common among multilingual societies. Up to 314.9 million Indians, according to research, speak many languages.

This causes a problem with code switching and mingling, particularly while using social networking sites [7][8]. Code mixing is the practice of using lexicons and syntax from many languages in a single statement [9][10]. The fundamental difficulty in resolving code mixed problems is the lack of adequately tagged dataset. [11]

"Emotion detection" is one of the most difficult topics in the field of natural language processing and is the subject of this paper.

In comparison to the English language, Hindi-English code-mixed texts are still mostly undiscovered and have not received as much attention [12]. This topic is addressed and a call for more research in the field is made using 150k tweets in a code-mixed Hindi-English dataset that has been annotated. By utilising bi-lingual self trained word embedding on a code mixed data, transformer based models like BERT, LSTM, Bi-LSTM, and CNN models, this study aims to examine a variety of deep learning models.

II RELATED WORKS

Due to the massive growth of microblogging websites, there have been an increase in interest in identifying moods, emotions in large text corpora [13][14]. Tests with text based emotion categorization in children fairy tales along the lines of fundamental emotions were undertaken in the first study on emotion recognition in textual data [15][16]. The authors of a companion article [17] look into real-world knowledge bases that emphasise people's inherent responses to a range of circumstances in order to recognise emotions at the phrase level. As non native English speakers use media more regularly, sentiment classification of regional languages on

social networks and coding mixed data are becoming increasingly common.

The Hindi corpus writers successfully retrieved sentiment lexons from HindiWordNet in a significant piece of work, and they attained an accuracy of 86% in the field of movies [18]. A thorough analysis of data collected from Facebook users who could converse in both English and Hindi revealed that 17.1% of all posts, or about one-fourth of all terms in their dataset, contained few indication of code mixing [19]. On a dataset including Hindi and English code, a sub-word level LSTM framework for sentiment classification was suggested [20]. A corpus that is code-mixed between Hindi and English was employed in experiments that included supervised learning (SVM) for emotion recognition [21].METHODOLOGY

The aspects of the suggested methodology, such as dataset collecting, labelling, data pre-processing, semantic similarity, and deep learning models, are described in detail in this section.

Dataset

2867 tweets are included in the publication's annotated dataset [21]. Using the TwitterScraper API, we generated a class-balanced collection of Hinglish data with pertinent search tags like #joyful, #unhappy, #frustrated, #worry, #despair, #amaze, etc. This information was insufficient for carrying out any useful machine learning work due to the issue of overfitting.

Emotion	Number of Instances
Happy	25870
Sad	20930
Anger	28710
Fear	18980
Disgust	35778
Surprise	18939
Total Sentences	149207

Table 1. Number of Tweets pe class

B. Dataset Analysis

2,50,000 tweets are scraped for analysis. Reducing of the data to a class proportioned corpora of 1,50,000 tweets after deleting the noisy cases that contained unrecognised characters was done. The tweets were labelled with six typical emotions [16]. The hashtags that were utilised as search parameters for scraping the tweets were used to annotate them. A cheerful label was placed on each case that was found using a hashtag, such as #yayy. This process was carried out for each of the six emotions under investigation. The total numbers of tweets by class are displayed in Table 1.

Some examples of labelled data

Tweet: Great darshann today at MahamahalakshmiTemple coupled with aarti! #joy @pratik2543 Translation: Had a wonderful experience at Mahamahalakshmi Temple, including the ceremonies. EMOTION: Happy

Tweet: Akele hi gujarti hai jindagi, log to keval tasalli dete hain, sath nahi. #Sad :(

Translation: Life passes alone, people only give comfort, not together. #Sad :(

Emotion: Sad

C. Dataset Pre-processing

Only Hinglish tweets remained after Devanagari and pure English tweets were purged from the scraped data. We also got rid of URLs, punctuation, references, "#" signs, unusual terms (words with fewer than ten occurrences in the overall data), and keywords from scraping in order to give our models cleaner data (such as joy, unhappy, etc.).

D. Word Embedding Methods

Because this is a text classification problem with numerous labels, the text must first be translated into a format that the machine learning algorithms can comprehend. Word embeddings are a means to quantitatively represent words. Particularly, word embeddings are vector representation of words that be learned unsupervisedly and whose relative similarity is proportional to their semantic similarity [22].

The issue is addressed by the use of two different word embedding types, each was trained on two separate dataset after preprocessing (removing hashtag, username mentions, Web links, grammar, and scraping-related words), one of which contained only Hindi-English tweets and the other a mixture of English and Hindi tweets. To find the proper correlation between the lexicons of the two languages, then combined tweets in Hinglish and English.

FastText: FastText, a Facebook feature, took the place of Word2Vec embeddings in 2016. Unlike word2vec, which gives the network single words, FastText thinks a word is made up of character n-grams [23]. As a result, a given word is divided up into several sub-words (Example: today, to, ay, day, da). In addition to learning the weights for the entire word, a FastText model must also learn the weights for each letter's n-gram. Because it is now highly likely that parts of their ngrams are contained in these other word, it may approximate unusual words and represent words that are not in the corpus, in contrast to Word2vec.

E. Deep Learning Algorithms

There are six deep learning based methods used to tackle the problem of emotion recognition in text-coded data. Many models have been proposed to analyse the feelings of the hinglish tweets dataset, including CNN, LSTM, and Bi-LSTM and BERT. We used two distinct sets of data to train FastText word representations: one kind contained online Hindi-English text, and the other contained both Hindi and English text. These embeddings were used as input by all of the suggested models, with the exception of the transformer-based models, to predict the sentiment of the tweet.

1) Convolutional Neural Network. (CNN)

When given inputs of images for multiclass classification problems, CNN has been demonstrated to perform effectively [24]. In this instance, word embeddings are used as inputs, and before conducting the final classification, characteristics are extracted. Figure 1 shows the network layout that was used. A few selected word vector representations from the tweet with consideration are transferred to the model structure by the embedding layer, which is the first layer. The output of the embedding layer is fed after a global max pooling layer, dropout, and four convolutional layers. The final layer, which serves as the categorization layer, is the most dense and connected of the following three levels. Dropout was used to increase convergence and close the accuracy gap between training and validation.

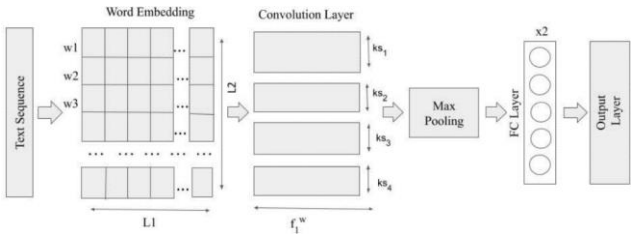

Figure 1. CNN Architecture

2) RNN

A word meaning depends on the context in which it is used, which can have a big impact on how the sentence is perceived as a whole.

Sentence 1: This rock is very hard

Sentence 2: He is very talented and hardworking person.

The word "hard" is employed in two different ways in these remarks, and as a result, the word can mean different things depending on the circumstance. Because they have unique ways of extracting word context from its surrounds, RNNs are helping in the modelling word context.

3) LSTM: It has been demonstrated that LSTMs can handle the issue of vanishing gradients and store the necessary context for words [25]. [26] A word's context is determined by the words that come before it. The meaning of words that have already been used is stored in memory cells that are incorporated into the network by LSTMs. An LSTM-based network is built to simulate this scenario.

4) Bi- LSTM: It has been successfully included into text classification applications to extract context [27]. The words that follow before it and after it help to establish the context of a word. Neural connections in both directions are necessary for the brain to recall the words that occur before and after a specific phrase..

5) Bi- LSTM with Attention: This method of paying attention relies on recognising the words that have the biggest impacts on the sentence's overall emotional content and removing the ones that are merely "noise" or inconsequential additions. There are variations in how states are integrated and sent to the fully connected (FC) layers in attention based BiLSTM.

6) BERT (bert-base-uncased): [28] It uses a combination of objectives designed for new sentence prediction, masked modelling tasks. It is a bidirectional based transformer model that was pretrained on a sizable Wikipedia and Toronto book corpus.

III Experimental Environment

With a 10% split over the total training dataset, the model underwent 20 iterations of training. Specifically, we used the checkpoint that was saved right before the model began to overfit to compute the metrics on the 10% test dataset split. We stored the model checkpoints at each epoch.

Many hyperparameters are used to train models and embeddings, including different activation functions, loss functions, and optimizers.

The Adam optimizer with categorical cross-entropy loss loss function produced the best results across all of the deep learning models described. With the exception of the output layer, which features sigmoid activation function, layers used ReLU activation.

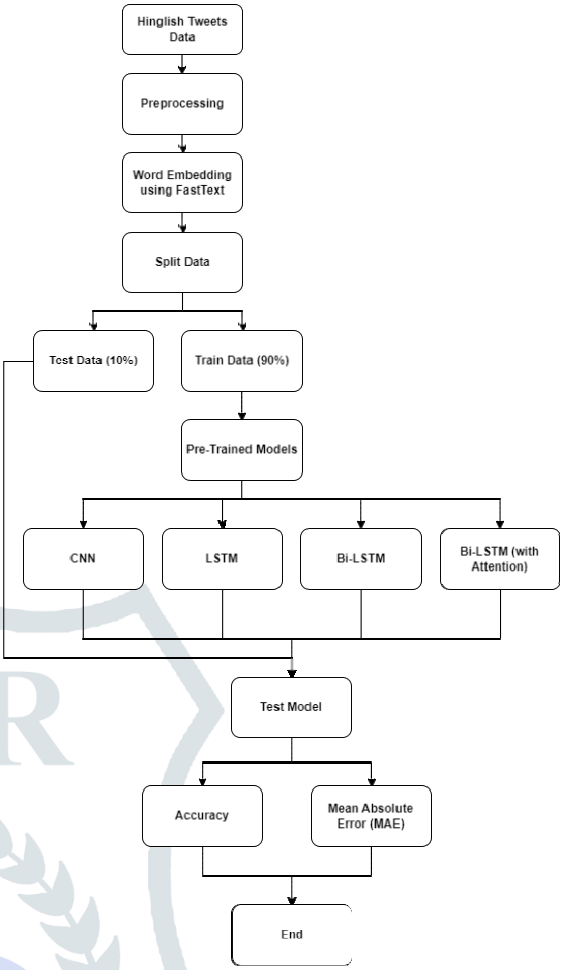

Fig 2. Overview to Sentiment Analysis

V Result

DL Models	Accuracy	Val Accuracy	MAE (h)
CNN	75.25	62.33	0.6959
LSTM	66.80	64.42	0.9069
Bi-LSTM	71.27	67.50	0.7982
Bi-LSTM (Attention)	73.25	87.78	0.7381
BERT	72.21	81.43	0.7421

Table 1. Deep Learning Models Accuracy

The SVM classifier model displayed an accuracy of 58% while working with the same sentiment labels as in the prior study, i.e., the baseline model [21]. Table 1 of this paper's suggested deep learning models shows that CNN fared best overall in terms of accuracy and mean absolute error (MAE), while Bi-LSTM with attention performed best in terms of validation accuracy for the Hinglish Code Mixed dataset.

Addressing language issues complexity connected with code mixed data and the lack of clean data are the main barriers to the approach of identifying emotions in Hindi English code mixed data. In order to reduce the impact of noise, which includes spelling mistakes, stem words, and the presence of numerous contexts, it is necessary to require even more tag-based cleaner data.

IV Conclusion

Emotion detection and opinion mining have emerged as crucial study areas as a result of the surge in recent years in the use of social media for the unrestricted expression of attitude and opinion.

The performance of several deep learning algorithms under text embedding carried out using FastText process is tested in this paper using labelled Hinglish code mixed data, with tweets scraped using TweetScrapper API representing various emotional states, such as joy, unhappy, frustrated, worry, disgust, and amaze. Convolutional Neural Network performed best among the suggested models, providing an accuracy of 75.25 percent. For further work, sentiment analysis can also be performed by contrasting suggested deep learning models with other transformer models.

V References

[1] Jean Greaves, Travis Bradberry, and Patrick M. Lencioni. 2009. Emotional Intelligence 2.0. CA : TalentSmart, San Diego.

[2] Maximilian Schmitt, Fabien Ringeval, and Bjorn W. ¨ Schuller. 2016. At the border of acoustics and linguistics: Bag-of-audio-words for the recognition of emotions in speech. In INTERSPEECH, pages 495– 499.

[3] Byoung Chul Ko. 2018. A brief review of facial emotion recognition based on visual information. Sensors, 18(2):1–20.

[4] Rosalind W Picard. 2000. Affective computing. MIT press.

[5] Ying Chen, Sophia Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. volume 2, pages 179–187.

[6] Lea Canales and Patricio Mart´ınez-Barco. 2014. Emotion detection from text: A survey.

[7] Sakshi Gupta, Piyush Bansal, and Radhika Mamidi. 2016. Resource creation for hindi-english code mixed social media text.

[8] Stella Monica, M ´ onica C ´ ardenas-Claros, and Neny ´ Isharyanti. 2009. Code switching and code mixing in internet chating: betwen ”yes”, ”ya”, and ”si” a case study. The jaltcall Journal, Vol 5:67–78.

[9] Shana Poplack and James Walker. 2003. Pieter muysken, bilingual speech: a typology of codemixing. cambridge: Cambridge university press, 2000. pp. xvi+306. Journal of Linguistics, 39:678 – 683.

[10] Peter Auer and Li Wei. 2007. Handbook of Multilingualism and Multilingual Communication. De Gruyter Mouton, Berlin, Boston.

[11] Dong-Phuong Nguyen and A. Seza Dogruoz. 2013. Word level language identification in online multilingual communication. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 857–862, United States. Association for Computational Linguistics (ACL).

[12] Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In Proceedings of the 10th International Conference on Text, Speech and Dialogue, TSD'07, page 196–205, Berlin, Heidelberg. Springer-Verlag.

[13] Efthymios Kouloumpis, Theresa Wilson, and Johanna D. Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In ICWSM, pages 538–541.

[14] Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. volume 10.

[15] Cecilia Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. pages 579—-586.

[16] Paul Ekman. 1992. An argument for basic emotions. Cognition and Emotion, 6(3-4):169–200.

[17] Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Corpus creation and emotion prediction for Hindi-English code-mixed social media text. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 128–135, New Orleans, Louisiana, USA. Association for Computational Linguistics.

[18] Amit Mandelbaum and Adi Shalev. 2016. Word embeddings and their use in sentence classification tasks.

[19] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

[20] Wael Ezat, Mohamed Dessouky, and Nabil Ismail. 2020. Multi-class image classification using deep learning algorithm. Journal of Physics: Conference Series, 1447:012021.

[21] Duc Tran, Hieu Mac, Van Tong, Hai-Anh Tran, and Giang Nguyen. 2017. A lstm based framework for handling multiclass imbalance in dga botnet detection. Neurocomputing, 275.

[22] Sepp Hochreiter and Jurgen Schmidhuber. 1997. ¨ Long short-term memory. Neural Comput., 9(8):1735–1780.

[23] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspectlevel sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 606–615, Austin, Texas. Association for Computational Linguistics.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.