Synergizing CNN and Self-Attention based LSTM for Analyzing Sentiments in Code-Mixed Text

Yasar Arafat E
Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology
Chennai, India
Email: **yasararafate@gmail.com**

Arjun A
Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology
Chennai, India
Email: **umaarjun2002@gmail.com**

Dr. A. Viji Amutha Mary
Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology
Chennai, India
Email: **vijiamuthamary.cse@sathyabama.ac.in**

***Abstract*— This paper examines the various applications of sentiment analyses in code-mixed text, from annotating reviews by users to identifying political or societal sentiments among certain subgroups. We propose an ensemble architecture to conduct sentiment analysis of code-mixed tweets. The deep learning architecture of this project combines deep-learning technologies such as CNN or self-attention-based LSTM neural networks. CNN is a key component in our architecture, allowing us to differentiate between positive and negatively-toned tweets. Convolutional layers are excellent at identifying features in text documents and accurately classifying sentiments for this polarized statement. The LSTM (Long-Term Memory), neural network component for Neural tweets, has the ability to distinguish the correct sentiment in texts that include multiple units that express that sentiment. It can also navigate code-mixed text that contains mixed sentiments and accurately classify tweets as neural Tweets.**

Keywords—deep learning, sentimental analysis, code-mixed text, CNN, LSTM.

I. INTRODUCTION

The use of code mixing is common in many forms of communication, from social media to written texts. Although code-mixed languages include words from various linguistic sources, we will focus on bilingual code mix as it relates to Hinglish or Spanglish in written texts. The practice of categorizing human emotions and affections is called sentiment analysis. Understanding Sentiment Analysis requires an understanding of Human Emotions and Affection States as a base for decision-making. It is not easy to code-mix languages, as sentences may not match a specific language model. As tokens, hashtags and usernames are often used in mixed text, hashtags or usernames may be used as tokens and the XLMR Embeddings are then used to build both CNN-LSTM ensembles as well as self-attention LSTM ensembles.

The deep learning network architectures like Convolutional Neural Networks have been extensively used in previous studies to analyze sentiment. However, none of these efforts included a CNN and a model that relied on self-attention. Our research showed this combination is highly effective for handling positive and negative tweets, analyzed using CNN segments, while the self-attention component was more adept at handling those neutral ones.

This paper presents a combined concept architecture of neural networks and self-attention-based LSTM models using XLM-RoBERTa Encoder. The ensemble approach combines both components to achieve optimal performance.

II. Related Work

The modelling of deep learning and NLP tasks using mixed data poses many challenges. Ali Athar. (2020) I am trying to find an POS labelling system for mixed products.

Nazanin's other works. (2021) used CNN for sub-word embeddings, and then embedded the embeddings into a two-dimensional neural network LSTM to learn sub word information from SNS on the Internet. Altwords are popular texts with incorrect punctuation and spelling.

This function does not provide semantic information at the word-level. This allows for a more detailed examination of the effects of words embedding-based techniques. Aqsa's new work. They used bilateral LSTM networks, and an additional feature net which they called common and unique encoders. This approach integrates a neural net using a tracking method that allows all views to be evaluated using a weighted-approach while presenting a mixture of local views.

III. Proposed Approach

A. Preliminary Data Processing

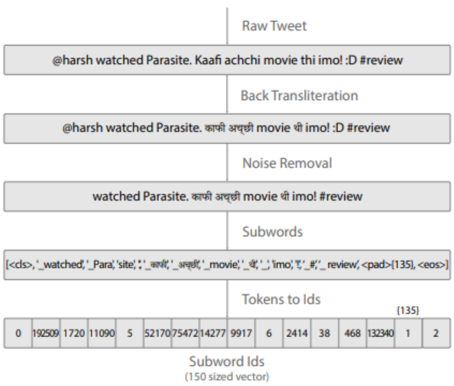

Figure 1: Preprocessing steps

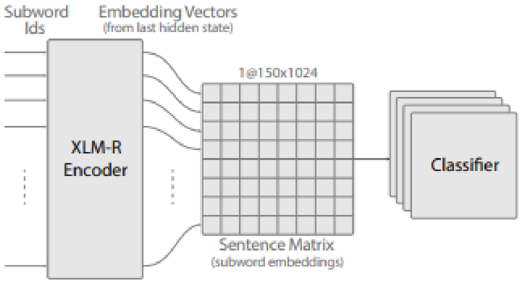

Figure 2: XLM-RoBERTa

Initially the data scraped from the Internet have to be processed. The following pre-processing is used to prepare tweets for the training phase (Figure 1).

1. Translation reversed: All words that contain "Hindi" are translated into Sanskrit in all languages. Google's Transliteration API2 will be used. Messages written in "Spanish", however, will not be translated.

2. Noise reduction: The hashtag, URL and emoticons (annotated with #hashtag), as well as the user's username (annotated @username) are removed. The emojis were replaced with their corresponding content. However, removing them would result in better performance.

3. Tokenization: Tweets that have been filtered for noise are tagged with sub-words using the XLM R (Conneaut et al., 2018) language, and then converted to ID letters.

B. Embedding layer

It is important to use multiple languages, as our database contains tweets and numbers. Our proposed design uses XLM R embeddings. XLM is a tool for language transformation that uses more than 2 TB SNS filter data from the Internet to know the mask model. The sub-word IDs are sent into the XLM-RoBERTa encoder.

Accessing the distribution function requires looking at each token's last known state, or "hidden last state". Training phase. In this step, the XLM coder was tuned for more accurate encoding while Devlin et al's

BERT multilingual was evaluated as part of this phase. We also tested the Multilingual BERT (hereafter called M-BERT), published by Devlin et al. We also tested Devlin et al.'s multilingual BERT. (2018). We have experimented and come to conclusion that RoBERTa performs better on the two datasets.

IV. Architecture

We propose a deep learning ensemble model consisting of two major components in the architecture.

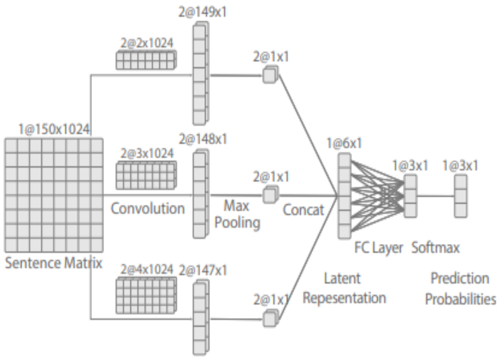

Figure 3: CNN Classifier

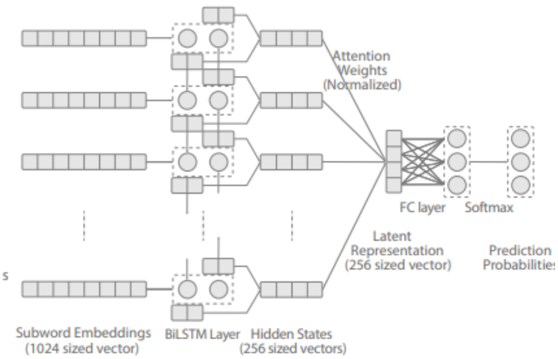

Figure 4: Self-Attention Classifier

A. Convolutional Neural Network

The first component is a Convolutional Neural Network. CNN takes into account both context and word order when considering each word that appears, while embeddings are created by passing each sub word of a sentence sequentially into dimensions. Convolutional processes were then conducted using three distinct filter sizes (2"/3"/4") before biasing and activating RELU nonlinear algorithm.

Multiple filter sizes are utilized to capture content with differing lengths. Convolution extracts features from word windows while max pooling extracts features that are important to the feature map. Furthermore, an XLM embedding element receives this message, then sends SoftMax features with which it produces SoftMax predictions known as pCNN predictions.

This element receives the XLM-R embedding and produces the prediction through the SoftMax function. These predictions are denoted as pCNN.

B. Self-Attention based LSTM

Second Component is an auto-attentive classifier system. When faced with mixed emotions, this classification aids in ascertaining what the general sentiment might be. Soft colour (Xu et al., 2015) represents a decision involving different colours, where SoftMax assigns weights to each word, and the color results indicate the latent weight of each field's representation.

Hochreiter and Schmid Huber's (1997) Bidirectional Short-Term Memory Bi-LSTM employing output from an XLM R encoder as its initial layer of storage is called Bi-LSTM. The hidden states of Bi-LSTM are used to derive the efficiency of the model. In the segment of tokenized words that we have created, we represent the hidden state in the Bi-LSTM in the forward hidden state and the backward hidden state.

The compound description ki is obtained by combining forward hidden state and the backward hidden state. Combining the forward and reverse hidden states results in the expression combination (k1,k2,...,kn).

The representation is the combined weight of all hidden states. Subsequently, the representation vector h undergoes a fully connected process, and SoftMax is applied to obtain the estimated pattern. Element-by-element (refer Figure 5) (indicated by ○) are employed to combine the predictions from the first and second components, resulting in the final prediction. Other aggregation methods were explored, and the element-wise combination proved to be more effective.

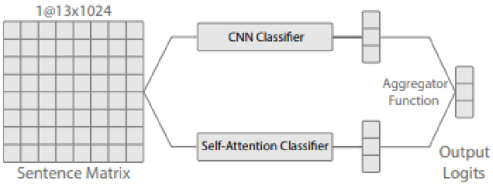

Figure 5: Classifier

Table 1: Training, Validation and Test data

	Dataset	Positive	Neutral	Negative
Train	Hinglish	5264	4634	4102
	Spanglish	6005	3974	2023
Validation	Hinglish	982	1128	890
	Spanglish	1498	994	506
Test	Hinglish	1000	1100	900

V. Dataset

The data model is created by scraping data from SNS platforms. We use this to train Indian English and Spanish English patterns. The data set is recorded semi-automatically. Table 1 provides training, validation, and testing data for the database. The Hinglish state is balanced, while the Spanglish state is unbalanced. Use the verification process provided by your organization for hyperparameter tuning.

VI. Results of the experiments

The self-monitoring system outperformed previous models in terms of average data. It also performed better on both positive and negative models.

It is believed that tweets are effective because they can evoke a variety of emotions. The combination of CNN results and Self-Monitoring Models outperformed previous models.

It achieved a return score of 0.717, and an F1 of 0.720 on the first dataset, and a score of 0.712 with an F1 of 702 on the second dataset. In Figures 6 and 7, confusion matrices (o: fair; +: good; -: poor) are displayed for clusters in the two datasets.

Our project showed a state-of-the-art result with the most accuracy than any other models currently.

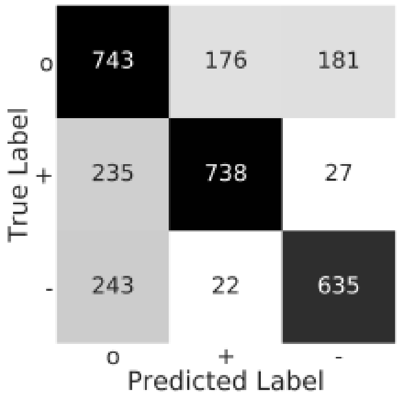

Figure 6: Hindi-English dataset Confusion matrix

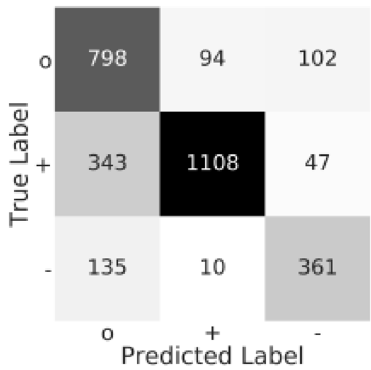

Figure 7: English-Spanish dataset Confusion matrix

Table 2: Results of the model on the two datasets

	F1				Macro	Macro
	o	**+**	**-**	**Macro**	**Precision**	**Recall**
Hinglish	0.640	0.762	0.729	0.707	0.712	0.705
Spanglish	0.135	0.825	0.375	0.725	0.763	0.696

VII. Analyzing the results

A. Sentence Vectors Result Visualization

The t-SNE method was used to create a visual model of sentence embeddings for the Hinglish data (van der Maten & Hinton, 2008). The two points in Figure 8 are where all sentence vectors were projected. CNN grouped tweets into two distinct groups: positive and negative. Neutral tweets were dispersed among the two. In contrast, when the Data is neutral the points are partially dispersed. These two concepts are complementary, and can improve predictions for all groups.

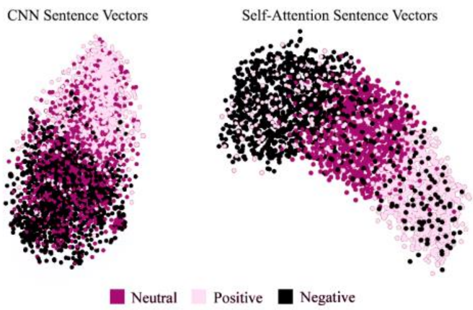

Figure 8: Sentence Vectors

B. Error Analysis

The majority of errors involve tweets that fall into three categories:

1. Neutral - Although self-care has improved the performance of tweets, neutral tweets are still behind both +ve and -ve sentiments.
2. Mocking and witty comments are generally posted on sns by the users. It was difficult to classify. Due to their secret histories, tweets that made a good impression on fake atheists were hard to classify. This led to inaccurate estimations of how good they were.
3. Minor Negatives - Because of the large number malicious tweets that were in the database, certain negative tweets such as "Ek Gaun mein ek Kisan is Dead" were incorrectly rated as average.

VIII. Conclusion

We used a set CNNs with self-developed architectures, and XLM R multilingual embeddings for our system. We evaluated the performance of various models for different groups of tweets. LSTM proved useful in classifying neutral data which is complex and hard to understand due to many emotional and behavioural factors. CNN was incorporated for holistic integration to help identify all three groups. Our model's performance was visualized using the t-SNE algorithms, which showed improvement over previous studies.

References

[1] Supriya Sameer Nalawade, Arun S. Patil, “An Empirical Study on Sentimental Analysis using Deep Learning”, 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC).

[2] Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang, “Dimensional sentiment analysis using a regional CNN-LSTM model”, 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 225–230, Berlin, Germany, August. Association for Computational Linguistics.

[3] Prabha PM Surya, B Subbulakshmi, “Sentimental Analysis using Naive Bayes Classifier”, 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN).

[4] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio, “: Neural image caption generation with visual attention. ”, CoRR,abs/1502.03044.

[5] Phasit Charoenkwan, “ThaiFBDeep: A Sentimental Analysis Using Deep Learning Combined with Bag-of-Words Features on Thai Facebook Data”, 2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI).

[6] Joosung Yoon and Hyeoncheol Kim, “Multi-channel lexicon integrated CNN-BiLSTM models for sentiment analysis” 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017), pages 244–253, Taipei, Taiwan, November. The Association for Computational Linguistics and
Chinese Language Processing (ACLCLP)

[7] Abdulrahman Radaideh, Fikri Dweiri, Mohammad Obaidat, “A Novel Approach to Predict the Real Time Sentimental Analysis by Naive Bayes & RNN Algorithm during the COVID Pandemic in UAE”, 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI).

[8] Gazi Imtiyaz Ahmad, Jimmy Singla, “(LISACMT) Language Identification and Sentiment analysis of English-Urdu ‘code-mixed’ text using LSTM”, 2022 International Conference on Inventive Computation Technologies (ICICT).