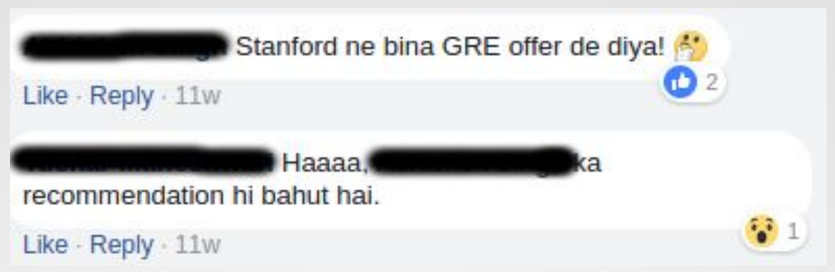
Consonant-Vowel Sequences as Subword Units for Code-Mixed Languages

Upendra Kumar, Vishal Singh Rana, Chris Andrew, Santhoshini Reddy and Amitava Das
Indian Institute of Information Technology (IIIT), Sri City, AP, India, 517541
{upendra.k14, vishal.s14,chris.g14, santhoshini.r14, amitava.das}@iiits.in

1 Social Media and Code-Mixed Languages



2 Objective

- In social media, non-English speakers [according to statistics half of messages on Twitter aren’t in English (Schroeder, Minocha, and Schneider 2010)[4]] do not always use English to express their thoughts.
- They use mixed languages to express their thoughts. This phenomenon is called code-mixing.
- Our work focuses on addressing the specific challenges of using out-of-vocabulary words required for developing an efficient model to analyze sentiment from Hindi-English code-mixed language.
- Introduces new phonemic sub-word units for Hindi-English code-mixed text along with a hierarchical deep learning model.
- In order to do so efficiently, we introduce a heuristic for segmenting words in phonemic sub-word units instead of using word or character level features.

3 Related Work

- (Sharma, Srinivas, and Balabantaray 2015)[5] use Hindi SentiWordNet and normalization techniques to detect sentiment in Hi-En code-mixed tweets.
- (Rudra et al. 2016)[3] use lexicon based features (swear words, exclamation marks, sentiment words and negation words) for developing a classifier for sentiment analysis.
- A deep learning based method for Hindi-English code-mixed text was proposed by (Joshi et al. 2016). For Hi-En code-mixed text (Joshi et al. 2016)[3] address the problem of rare or out-of-vocabulary words without any text normalization.

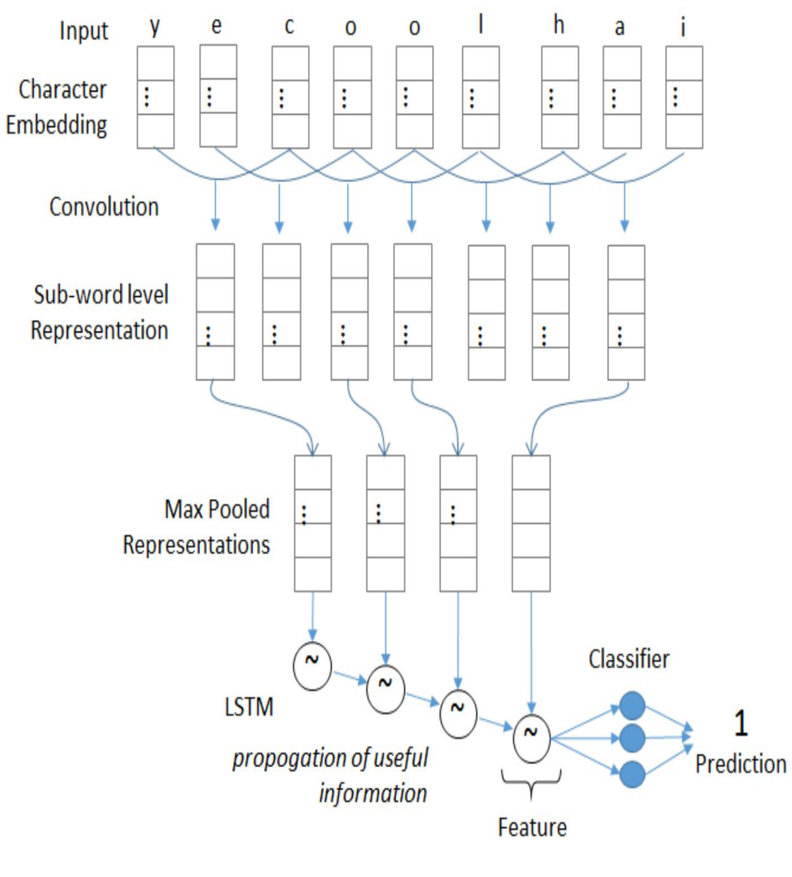


Figure 1: A CNN-LSTM model using character n-grams as sub-word units

- In this paper, a novel approach is proposed without any need of explicit text normalization by creating sub-word units and a new hierarchical model that efficiently learns sentence representations from these units.

3 Dataset

- A dataset of 18K tweets was created using the Twitter API by querying tweets from Twitter accounts that frequently use Hindi-English code-mixed style for tweeting.
- Based on the sentiment of the text, data was manually annotated into three classes: positive, neutral and negative (-1, 0,1).

	EN	HI	MIX
Positive	9.93	14.8	3.09
Negative	5.24	15.9	1.60
Neutral	14.94	29.98	4.27

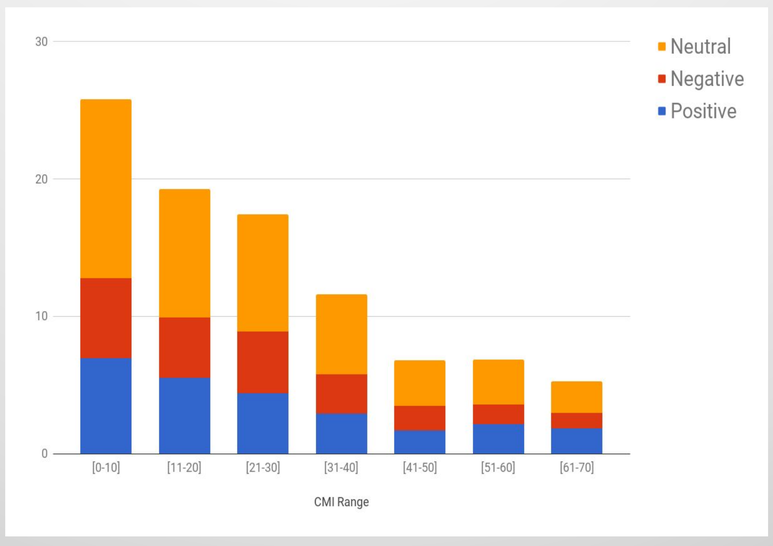


Figure x: Class distributions for different levels of Code Mixed Index for the proposed dataset.

4 Methodology

- The use of Roman script for a Hindi word may produce spelling variations, as the Roman script is not an Abugida script.
- Abugida scripts have a one-to-one mapping between spelling and pronunciation, where the same units will have the same pronunciations regardless of their context.
- The variations in spelling produce a number of rare and out-of-vocabulary words that introduce errors in the classification process.

Word	Meaning	Variations
gussa	*angry*	*gusa, gussaa*
kahan	*where*	*kaha, kahaan*
bahut	*very*	*bahout, bahot, bhot*

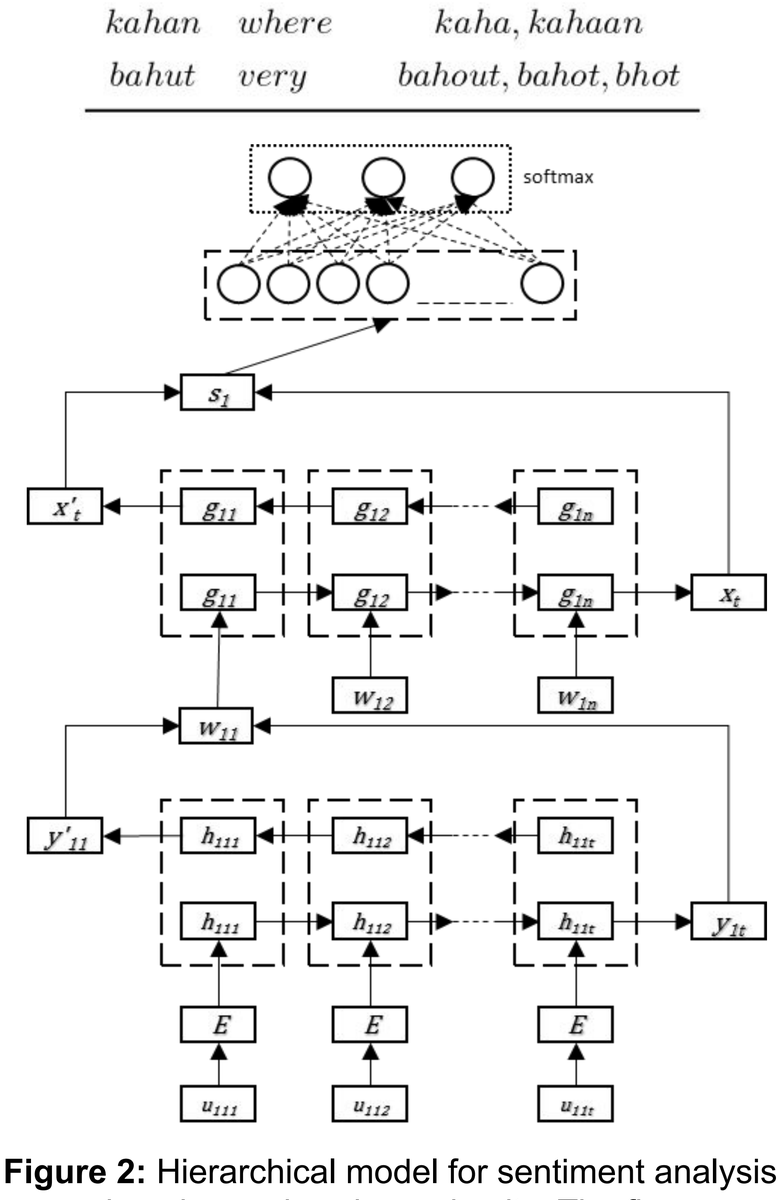


Figure 2: Hierarchical model for sentiment analysis using phonemic sub-word units. The figure illustrates the graph for sentence s 1 .

- To eliminate the problems associated with rare words we segment them into more commonly occurring phonemic units.
- Words are segmented into sub-word sequences of C^+V^+ that acts as an approximate syllable consisting of onset and rime (without coda) parts.
- Each sentence s_i is first tokenized into word tokens $w_{i,j}$. For each word token $w_{i,j}$, a list of approximate syllables/sub-word units $u_{i,j,k}$ is obtained. In order to obtain the predicted sentiment y_i for sentence s_i, we propose a hierarchical BiLSTM network using these units as highlighted in Figure 2.
- The first layer corresponds to the embedding layer which encodes the sequences of sub-words to their corresponding vector representations.
- The second layer functions as a word encoder which learns to compose representations of constituent sub-word units $u_{i,j,k}$ into representation of a word $w_{i,j}$.
- The third layer obtains representation for a sentence s_i.
- Finally, a fully connected layer is added to use these representations in order to infer and predict the sentiment associated with the sentence.

Example

- **Kitab** : ki + ta + b
- **Book** : boo + k
- **Chandrama** : Chan + dra + ma

5 Experiments and Results

- In order to validate our model, multiple experiments were done using different word based model architectures using LSTM and CNN as shown in Table 1. For experiments involving CNNs two convolutional layers with kernels of size 3 and 4 were used.
- The next set of experiments consisted of a single LSTM trained using various embeddings. The best performance of word-based classifier was 50.43% using LSTM with pre-trained GloVE embeddings.
- Further, the sub-word LSTM model proposed in [2] was used for the purpose of comparison. It performed better than other word-based models yielding 57.88% accuracy.
- In next experiment, we used the proposed sub-word units with a hierarchical model as illustrated in Figure 2 yielding an accuracy of 74.62%.
- It is clear that the model proposed in this paper significantly outperforms the latter sub-word model by a large margin of 16.74%.
- It may be reasoned that the proposed sub-word units are on average more frequently co-occurring than n-grams and thus provide rich contextual information relevant to the classification task.

Method	Embeddings	Accuracy	F1-Score
Char LSTM[2]	Random	52.36%	0.5170
Sub LSTM[2]	Random	57.88%	0.5561
CNN	Glove	47.23%	0.4591
CNN	Google	49.41%	0.4689
CNN	W2V Code-mixed	45.7%	0.4312
LSTM	Glove	50.43%	0.534
LSTM	Google	48.21%	0.4612
LSTM	W2V Code-mixed	42.23%	0.4146
Proposed Model	Random	**74.62%**	**0.756**

Paper ID: 261