Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text

Aditya Joshi[1]*, Ameya Prabhu†[2]*, Manish Shrivastava[3] and Vasudeva Varma[1]
[1]Search and Information Extraction Lab
[2]Centre for Visual Information Technology
[3]Language Technologies Research Center
International Institute of Information Technology, Hyderabad (India)
{aditya.joshi, ameya.prabhu} @research.iiit.ac.in
{m.shrivastava, vv} @.iiit.ac.in

Abstract

Sentiment analysis (SA) using code-mixed data from social media has several applications in opinion mining ranging from customer satisfaction to social campaign analysis in multilingual societies. Advances in this area are impeded by the lack of a suitable annotated dataset. We introduce a Hindi-English (Hi-En) code-mixed dataset for sentiment analysis and perform empirical analysis comparing the suitability and performance of various state-of-the-art SA methods in social media.

In this paper, we introduce learning sub-word level representations in LSTM (Subword-LSTM) architecture instead of character-level or word-level representations. This linguistic prior in our architecture enables us to learn the information about sentiment value of important morphemes. This also seems to work well in highly noisy text containing misspellings as shown in our experiments which is demonstrated in morpheme-level feature maps learned by our model. Also, we hypothesize that encoding this linguistic prior in the Subword-LSTM architecture leads to the superior performance. Our system attains accuracy 4-5% greater than traditional approaches on our dataset, and also outperforms the available system for sentiment analysis in Hi-En code-mixed text by 18%.

1 Introduction

Code Mixing is a natural phenomenon of embedding linguistic units such as phrases, words or morphemes of one language into an utterance of another (Muysken, 2000; Duran, 1994; Gysels, 1992). Code-mixing is widely observed in multilingual societies like India, which has 22 official languages most popular of which are Hindi and English. With over 375 million Indian population online, usage of Hindi has been steadily increasing on the internet.

This opens up tremendous potential for research in sentiment and opinion analysis community for studying trends, reviews, events, human behaviour as well as linguistic analysis. Most of the current research works have involved sentiment polarity detection (Feldman, 2013; Liu, 2012; Pang and Lee, 2008) where the aim is to identify whether a given sentence or document is (usually) positive, negative or neutral. Due to availability of large-scale monolingual corpora, resources and widespread use of the language, English has attracted the most attention.

Seminal work in sentiment analysis of Hindi text was done by Joshi et al. (2010) in which the authors built three step fallback model based on classification, machine translation and sentiment lexicons. They also observed that their system performed best with unigram features without stemming. Bakliwal et al. (2012) generated a sentiment lexicon for Hindi and validated the results on translated form of Amazon Product Dataset Blitzer et al. (2007). Das and Bandyopadhyay (2010) created Hindi SentiWordNet, a sentiment lexicon for Hindi.

* indicates these authors contributed equally to this work.

†Corresponding Author

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 2482–2491, Osaka, Japan, December 11-17 2016.

Sentence variations
Trailer dhannnsu hai bhai
Dhannnsu trailer hai bhai
Bhai trailer dhannnsu hai
Bhai dhannnsu trailer hai

Table 1: Illustration of free structure present in code mixed text. All sentences convey the same meaning.

Word	Meaning	Appearing Variations
बहुत (bahut)	very	bahout bohut bhout bauhat bohot bahut bhaut bahot bhot
मुबारक (mubaarak)	wishes	mobarak mubarak mubark
प्यार (pyaar)	love	pyaar peyar pyara piyar pyr piyaar pyar

Table 2: Spelling variations of *romanized* words in our Hi-En code-mix dataset.

Sentiment Analysis in Code-mixed languages has recently started gaining interest owing to the rising amount of non-English speaking users. Sharma et al. (2015) segregated Hindi and English words and calculated final sentiment score by lexicon lookup in respective sentient dictionaries.

Hindi-English (Hi-En) code mixing allows ease-of-communication among speakers by providing a much wider variety of phrases and expressions. A common form of code mixing is called as *romanization* [1], which refers to the conversion of writing from a different writing system to the Roman script. But this freedom makes the task for developing NLP tools more difficult, highlighted by (Chittaranjan et al., 2014; Vyas et al., 2014; Barman et al., 2014). Initiatives have been taken by shared tasks (Sequiera et al., 2015; Solorio et al., 2014), however they do not cover the requirements for a sentiment analysis system.

Deep learning based approaches (Zhang and LeCun, 2015; Socher et al., 2013) have been demonstrated to solve various NLP tasks. We believe these can provide solution to code-mixed and romanized text from various demographics in India, as similar trends are followed in many other Indian languages too. dos Santos and Zadrozny (2014) demonstrated applicability of character models for NLP tasks like POS tagging and Named Entity Recognition (dos Santos and Guimarães, 2015). LSTMs have been observed to outperform baselines for language modelling (Kim et al., 2015) and classification (Zhou et al., 2015). In a recent work, (Bojanowski et al., 2016) proposed a skip-gram based model in which each word is represented as a bag of character n-grams. The method produced improved results for languages with large vocabularies and rare words.

The *romanized* code mixed data on social media presents additional inherent challenges such as contractions like ”between” → ”btwn”, non-standard spellings such as ”cooolll” or ”bhut bdiya” and non-grammatical constructions like ”sir hlp plzz naa”. Hindi is phonetically typed while English (Roman script) doesn’t preserve phonetics in text. Thus, along with diverse sentence construction, words in Hindi can have diverse variations when written online, which leads to large amount of tokens, as illustrated in Table 2. Meanwhile there is a lack of a suitable dataset.

Our contributions in this paper are (i) Creation, annotation and analysis of a Hi-En code-mixed dataset for the sentiment analysis, (ii) Sub-word level representations that lead to better performance of LSTM networks compared to Character level LSTMs (iii) Experimental evaluation for suitability and evaluation of performance of various state-of-the-art techniques for the SA task, (iv) A preliminary investigation of embedding linguistic priors might be encoded for SA task by char-RNN architecture and the relation of architecture with linguistic priors, leading to the superior performance on this task.
Our paper is divided into the following sections:
We begin with an introduction to Code Mixing and romanization in Section 1. We mention the issues with code-mixed data in context of Sentiment Analysis and provides an overview of existing solutions. We then discusses the process of creation of the dataset and its features in Section 2. In Section 3, we introduce Sub-word level representation and explains how they are able to model morphemes along with propagating meaningful information, thus capturing sentiment in a sentence. Then in Section 4, we explain our experimental setup, describe the performance of proposed system and compare it with baselines and other methods, proceeded by a discussion on our results.

[1] https://en.wikipedia.org/wiki/Romanization

2 Dataset

We collected user comments from public Facebook pages popular in India. We chose pages of Salman Khan, a popular Indian actor with massive fan following, and Narendra Modi, the current Prime Minister of India. The pages have 31 million and 34 million facebook user likes respectively. These pages attract large variety of users from all across India and contain lot of comments to the original posts in code-mixed representations in varied sentiment polarities. We manually pre-processed the collected data to remove the comments that were not written in roman script, were longer than 50 words, or were complete English sentences. We also removed the comments that contained more than one sentence, as each sentence might have different sentiment polarity. Then, we proceeded to manual annotation of our dataset. The comments were annotated by two annotators in a 3-level polarity scale - positive, negative or neutral. Only the comments with same polarity marked by both the annotators are considered for the experiments. They agreed on the polarity of 3879 of 4981 (77%) sentences. The Cohen's Kappa coefficient (Cohen, 1960) was found to be 0.64. We studied the reasons for misalignment and found that causes typically were due to difference in perception of sentiments by individuals, different interpretations by them and sarcastic nature of some comments which is common in social media data. The dataset contains 15% negative, 50% neutral and 35% positive comments owing to the nature of conversations in the selected pages.

The dataset exhibits some of the major issues while dealing with code-mixed data like short sentences with unclear grammatical structure. Further, *romanization* of Hindi presents an additional set of complexities due to loss of phonetics and free ordering in sentence constructions as shown in Table 1. This leads to a number of variations of how words can be written. Table 2 contains some of the words with multiple spelling variations in our dataset, which is one of the major challenges to tackle in Hi-En code-mixed data.

Dataset	Size	# Vocab	Social	CM	Sentiment
STS-Test	498	2375	✓		✓
OMD	3238	6211	✓		✓
SemEval'13	13975	35709	✓		✓
IMDB	50000	5000			✓
(Vyas et al., 2014)	381	-	✓	✓	
Ours	3879	7549	✓	✓	✓

Table 3: Comparison with other datasets.

Popular related datasets are listed in Table 3. STS, SemEval, IMDB etc. have been explored for SA tasks but they contain text in English. The dataset used by Vyas et al. (2014) contains Hi-En Code Mixed text but doesn't contain sentiment polarity. We constructed a code mixed dataset with sentiment polarity annotations, and the size is comparable with several datasets. Table 4 shows some examples of sentences from our dataset. Here, we have phrases in Hindi (source language) written in English (target) language.

Example	Approximate Meaning	Sentiment Polarity
Aisa PM naa hua hai aur naa hee hoga	Neither there has been a PM like him, nor there will be	Positive
abe kutte tere se kon baat karega	Who would talk to you, dog?	Negative
Trailer dhannnsu hai bhai	Trailer is awesome, brother.	Positive

Table 4: Examples of Hi-En Code Mixed Comments from the dataset.

Our dataset and code is freely available for download [2] to encourage further exploration in this domain.

[2] https://github.com/DrImpossible/Sub-word-LSTM

3 Learning Compositionality

Our target is to perform sentiment analysis on the above presented dataset. Most commonly used statistical approaches learn word-level feature representations. We start our exploration for suitable algorithms from models having word-based representations.

3.1 Word-level models

Word2Vec(Mikolov et al., 2013) and Word-level RNNs (Word-RNNs) (thang Luong et al., 2013) have substantially contributed to development of new representations and their applications in NLP such as in Summarization (Cao et al., 2015) and Machine Translation (Cho et al., 2014). They are theoretically sound since language consists of inherently arbitrary mappings between ideas and words. Eg: The words person(English) and insaan(Hindi) do not share any priors in their construction and neither do their constructions have any relationship with the semantic concept of a person. Hence, popular approaches consider lexical units to be independent entities. However, operating on the lexical domain draws criticism since the finite vocabulary assumption; which states that models assume language has finite vocabulary but in contrast, people actively learn & understand new words all the time.

Excitingly, our dataset seems suited to validate some of these assumptions. In our dataset, vocabulary sizes are greater than the size of the dataset as shown in Table 3. Studies on similar datasets have shown strong correlation between number of comments and size of vocabulary (Saif et al., 2013). This rules out methods like Word2Vec, N-grams or Word-RNNs which inherently assume a small vocabulary in comparison to the data size. The finite vocabulary generally used to be a good approximation for English, but is no longer valid in our scenario. Due to the high sparsity of words themselves, it is not possible to learn useful word representations. This opens avenues to learn non-lexical representations, the most widely studied being character-level representations, which is discussed in the next section.

3.2 Character-level models

Character-level RNNs (Char-RNNs) have recently become popular, contributing to various tasks like (Kim et al., 2015). They do not have the limitation of vocabulary, hence can freely learn to generate new words. This freedom, in fact, is an issue: Language is composed of lexical units made by combining letters in some specific combinations, i.e. most of the combinations of letters do not make sense. The complexity arises because the mappings between meaning and its construction from characters is arbitrary. Character models may be apriori inappropriate models of language as characters individually do not usually provide semantic information. For example, while " $King - Man + Women = Queen$" is semantically interpretable by a human, "$Cat - C + B = Bat$" lacks any linguistic basis.

But, groups of characters may serve semantic functions. This is illustrated by $Un + Holy = Unholy$ or $Cat + s = Cats$ which is semantically interpretable by a human. Since sub-word level representations can generate meaningful lexical representations and individually carry semantic weight, we believe that sub-word level representations consisting composition of characters might allow generation of new lexical structures and serve as better linguistic units than characters.

3.3 Sub-word level representations

Lexicon based approaches for the SA task (Taboada et al., 2011; Sharma et al., 2015) perform a dictionary look up to obtain an individual score for words in a given sentence and combine these scores to get the sentiment polarity of a sentence. We however want to use intermediate sub-word feature representations learned by the filters during convolution operation. Unlike traditional approaches that add sentiment scores of individual words, we propagate relevant information with LSTM and compute final sentiment of the sentence as illustrated in Figure 1.

Hypothesis: We propose that incorporating sub-word level representations into the design of our models should result in better performance. This would also serve as a test scenario for the broader hypothesis proposed by Dyer et. al. in his impressive ICLR keynote [3] - Incorporating linguistic priors in network architectures lead to better performance of models.

[3] Available at: http://videolectures.net/iclr2016_dyer_model_architecture/

Methodology: We propose a method of generating sub-word level representations through 1-D convolutions on character inputs for a given sentence. Formally, let C be the set of characters and T be an set of input sentences. The sentence $s \in T$ is made up of a sequence of characters $[c_1, ..., c_l]$ where l is length of the input.

Hence, the representation of the input s is given by the matrix $Q \in R^{d \times l}$ where d is the dimensionality of character embedding that corresponding to $[c_1, ..., c_l]$. We perform convolution of Q with a filter $H \in R^{d \times m}$ of length m after which we add a bias and apply a non-linearity to obtain a feature map $f \in R^{l-m+1}$. Thus we can get sub-word level (morpheme-like) feature map. Specifically, the i^{th} element of f is given by:

$$f[i] = g((Q[:, i : i + m - 1] * H) + b) \tag{1}$$

where $Q[:, i : i + m - 1]$ is the matrix of $(i)^{th}$ to $(i + m - 1)^{th}$ character embedding and g corresponds to ReLU non-linearity.
Finally, we pool the maximal responses from p feature representations corresponding to selecting sub-word representations as:

$$y_i = max(f[p * (i : i + p - 1)]) \tag{2}$$

Next, we need to model the relationships between these features $y^i[:]$ in order to find the overall sentiment of the sentence. This is achieved by LSTM(Graves, 2013) which is suited to learning to propagate and 'remember' useful information, finally arriving at a sentiment vector representation from the inputs. We provide f_t as an input to the memory cell at time t. We then compute values of I_t - the input gate, $\tilde{C_t}$ - the candidate value for the state of the memory cell at time t and f_t - the activation of the forget gate, which can be used to compute the information stored in memory cell at time t. With the new state of memory cell C_t, we can compute the output feature representation by:

$$O_t = \sigma(Wy_t + Uh_(t - 1) + V(C_t + b) \tag{3}$$

$$h_t = O_t tanh(C_t) \tag{4}$$

where W,U and V are weight matrices and b_i are biases. After l steps, h_l represents the relevant information retained from the history. That is then passed to a fully connected layer which calculates the final sentiment polarity as illustrated in the Figure 1.

Figure 2 gives schematic overview of the architecture. We perform extensive experiments to qualitatively and quantitatively validate the above claims as explained in the next section.

4 Experiments

We perform extensive evaluation of various approaches, starting with a suitability study for the nature of approaches that would be able to generalize to this data. We compare our approaches with the state-of-the-art methods which are feasible to generalize on code-mixed data and (Sharma et al., 2015), the current state-of-the-art in Hi-En code-mixed SA task.

4.1 Method Suitability

Following approaches have been used for performing SA tasks in English but do not suit mix code setting:

- Approaches involving NLP tools: RNTN (Socher et al., 2013) etc which involve generation of parse trees which are not available for code mixed text;
- Word Embedding Based Approaches: Word2Vec, Word-RNN may not provide reliable embedding in situations with small amount of highly sparse dataset.
- Surface Feature engineering based approaches: Hashtags, User Mentions, Emoticons etc. may not exist in the data.

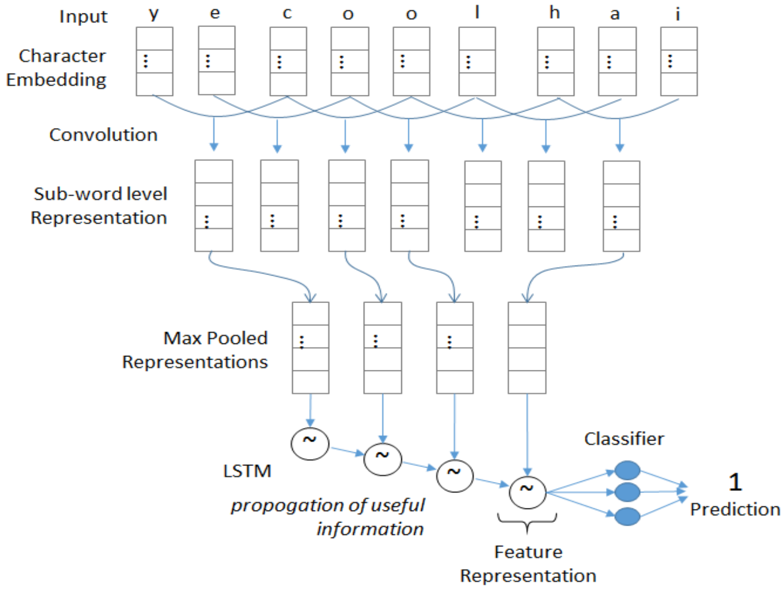


Figure 1: Illustration of the proposed methodology

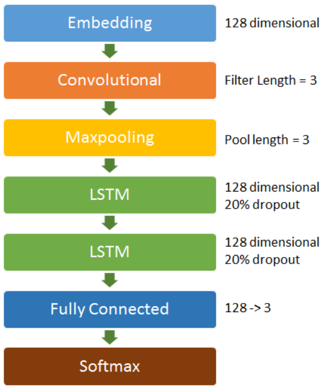


Figure 2: Schematic overview of the architecture.

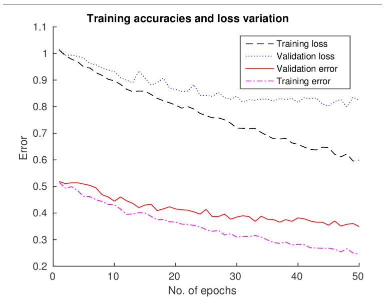


Figure 3: Training accuracy and loss variation.

4.2 Experimental Setup

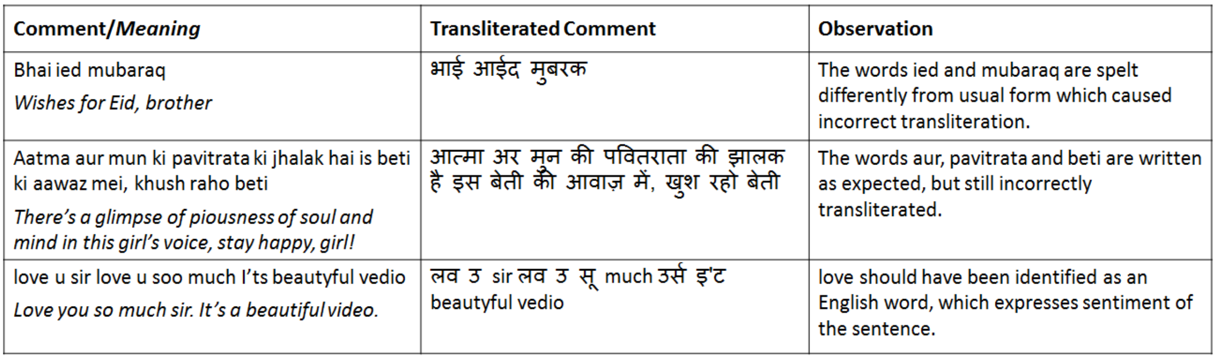
Our dataset is divided into 3 splits- Training, validation and testing. We first divide the data into randomized 80-20 train test split, then further randomly divide the training data into 80-20 split to get the final training, validation and testing data.

As the problem is relatively new, we compare state of the art sentiment analysis techniques (Wang and Manning, 2012; Pang and Lee, 2008) which are generalizable to our dataset. We also compare the results with system proposed by Sharma et al. (2015) on our dataset. As their system is not available publicly, we implemented it using language identification and transliteration using the tools provided by Bhat et al. (2015) for Hi-En Code Mixed data. The polarity of thus obtained tokens is computed from SentiWordNet (Esuli and Sebastiani, 2006) and Hindi SentiWordNet (Das and Bandyopadhyay, 2010) to obtain the polarity of words, which are then voted to get final polarity of the sentence.

The architecture of the proposed system (Subword-LSTM) is described in Figure 2. We compare it with a character-level LSTM (Char-LSTM) following the same architecture without the convolutional and maxpooling layers. We use Adamax (Kingma and Ba, 2014) (a variant of Adam based on infinity norm) optimizer to train this setup in an end-to-end fashion using batch size of 128. We use very simplistic architectures because of the constraint on the size of the dataset. As the datasets in this domain expand, we would like to scale up our approach to bigger architectures. The stability of training using this architecture is illustrated in Figure 3.

Method	**Reported In**	Our dataset		SemEval’ 13	
		Accuracy	**F1-Score**	**Accuracy**	**F1-Score**
NBSVM (Unigram)	(Wang and Manning, 2012)	59.15%	0.5335	57.89%	0.5369
NBSVM (Uni+Bigram)	(Wang and Manning, 2012)	62.5%	0.5375	51.33%	0.5566
MNB (Unigram)	(Wang and Manning, 2012)	66.75%	0.6143	58.41%	0.4689
MNB (Uni+Bigram)	(Wang and Manning, 2012)	66.36%	0.6046	58.4%	0.469
MNB (Tf-Idf)	(Wang and Manning, 2012)	63.53%	0.4783	57.82%	0.4196
SVM (Unigram)	(Pang and Lee, 2008)	57.6%	0.5232	57.6%	0.5232
SVM (Uni+Bigram)	(Pang and Lee, 2008)	52.96%	0.3773	52.9%	0.3773
Lexicon Lookup	(Sharma et al., 2015)	51.15%	0.252	N/A	N/A
Char-LSTM	Proposed	59.8%	0.511	46.6%	0.332
Subword-LSTM	Proposed	**69.7%**	**0.658**	**60.57%**	0.537

Table 5: Classification results show that the proposed system provides significant improvement over traditional and state of art method for Sentiment Analysis in Code Mixed Text



Comment/*Meaning*	**Transliterated Comment**	**Observation**
Bhai ied mubaraq *Wishes for Eid, brother*	भाई आईद मुबरक	The words ied and mubaraq are spelt differently from usual form which caused incorrect transliteration.
Aatma aur mun ki pavitrata ki jhalak hai is beti ki aawaz mei, khush raho beti *There’s a glimpse of piousness of soul and mind in this girl’s voice, stay happy, girl!*	आत्मा अर मुन की पवितराता की झालक है इस बेती कौ आवाज़ में, खुश रहो बेती	The words aur, pavitrata and beti are written as expected, but still incorrectly transliterated.
love u sir love u soo much I’ts beautyful vedio *Love you so much sir. It’s a beautiful video.*	लव उ sir लव उ सू much उर्स इ'ट beautyful vedio	love should have been identified as an English word, which expresses sentiment of the sentence.

Table 6: Output produced a by Hi-En Transliteration Tool

4.3 Observations

In the comparative study performed on our dataset, we observe that Multinomial Naive Bayes performs better than SVM(Pang and Lee, 2008) for snippets providing additional validation to this hypothesis given by Wang and Manning (2012).

We also observe that unigrams perform better than bigrams and Bag of words performs better than tf-idf in contrast to trends in English, as the approaches inducing more sparsity would yield to poorer results because our dataset is inherently very sparse. The lexicon lookup approach (Sharma et al., 2015) didn’t perform well owing to the heavily misspelt words in the text, which led to incorrect transliterations as shown in Table 6.

4.4 Validation of proposed hypothesis

We obtain preliminary validation for our hypothesis that incorporating sub-word level features instead of characters would lead to better performance. Our Subword-LSTM system provides an F-score of 0.658 for our dataset, which is significantly better than Char-LSTM which provides F-score of 0.511.

Since we do not have any other dataset in Hi-En code-mixed setting of comparable to other settings, we performed cross-validation of our hypothesis on SemEval’13 Twitter Sentiment Analysis dataset. We took the raw tweets character-by-character as an input for our model from the training set of 7800 tweets and test on the SemEval’13 development set provided containing 1368 tweets. The results are summarized in Table 5. In all the cases, the text was converted to lowercase and tokenized. No extra features or heuristics were used.

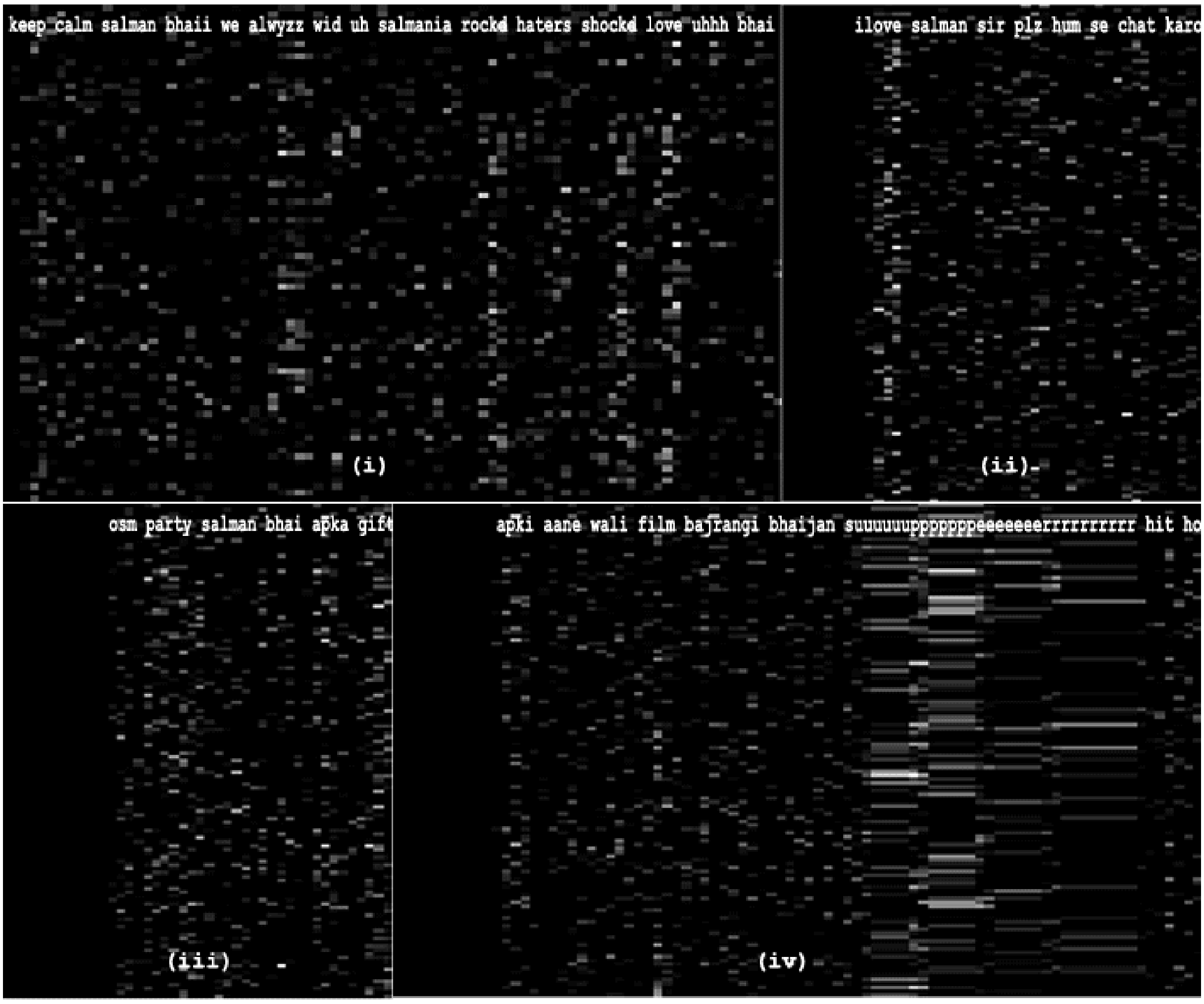


Figure 4: Visualization of the convolution layer for examples comments from the dataset show that word segments convey sentiment information despite being severely misspelt.

4.5 Visualizing character responses

Visualizations in Figure 4 shows how the proposed model is learning to identify sentiment lexicons. We see that different filters generally tend to learn mappings from different parts, interestingly showing shifting trends to the right which maybe due to LSTM picking their feature representation in future time steps. The words sections that convey sentiment polarity information are captured despite misspelling in example (i) and (ii). In example (iii), starting and ending phrases show high response which correspond to the sentiment conveying words (party and gift). The severe morpheme stretching in example (iv) also affects the sentiment polarity.

5 Conclusion

We introduce Sub-Word Long Short Term Memory model to learn sentiments in a noisy Hindi-English Code Mixed dataset. We discuss that due to the unavailability of NLP tools for Hi-En Code Mixed text and noisy nature of such data, several popular methods for Sentiment Analysis are not applicable. The solutions that involve unsupervised word representations would again fail due to sparsity in the dataset. Sub-Word LSTM interprets sentiment based on morpheme-like structures and the results thus produced are significantly better than baselines.

Further work should explore the effect of scaling of RNN and working with larger datasets on the results. In the new system, we would like to explore more deep neural network architectures that are able to capture sentiment in Code Mixed and other varieties of noisy data from the social web.

References

Akshat Bakliwal, Piyush Arora, and Vasudeva Varma. 2012. Hindi subjective lexicon: A lexical resource for hindi polarity classification. In *Proceedings of International Conference on Language Resources and Evaluation*.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code-mixing: A challenge for language identification in the language of social media. In *In Proceedings of the First Workshop on Computational Approaches to Code-Switching*.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *In ACL*, pages 187–205.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *AAAI*, pages 2153–2159.

Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identification using crf: Code-switching shared task report of msr india system. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 73–79.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, April.

Amitava Das and Sivaji Bandyopadhyay. 2010. Sentiwordnet for indian languages. In *Proceedings of the Eighth Workshop on Asian Language Resouces*.

Cícero Nogueira dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. *CoRR*, abs/1505.05008.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1818–1826.

Luisia Duran. 1994. Toward a better understanding of code switching and interlanguage in bilinguality: Implications for bilingual instruction. *Journal of Educational Issues of Language Minority Students*, 14:69–87.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, pages 417–422.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4):82–89, April.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Marjolein Gysels. 1992. French in urban lubumbashi swahili: Codeswitching, borrowing, or both. *Journal of Multilingual and Multicultural Development*, 13:41–55.

Aditya Joshi, Balamurali R., and Bhattacharyya Pushpak. 2010. A fall-back strategy for sentiment analysis in hindi: a case study. In *Proceedings of the 8th ICON*, Stroudsburg, PA. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-aware neural language models. *CoRR*, abs/1508.06615.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Pieter Muysken. 2000. *Bilingual Speech: A Typology of Code-mixing*. Cambridge University Press.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. 2013. Evaluation datasets for twitter sentiment analysis: A survey and a new dataset, the sts-gold. *1st Interantional Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013)*.

Royal Sequiera, Monojit Choudhury, Parth Gupta, Paolo Rosso, Shubham Kumar, Somnath Banerjee, Sudip Kumar Naskar, Sivaji Bandyopadhyay, Gokul Chittaranjan, Amitava Das, and Kunal Chakma. 2015. Overview of FIRE-2015 shared task on mixed script information retrieval. In Prasenjit Majumder, Mandar Mitra, Madhulika Agrawal, and Parth Mehta, editors, *Post Proceedings of the Workshops at the 7th Forum for Information Retrieval Evaluation, Gandhinagar, India, December 4-6, 2015.*, volume 1587 of *CEUR Workshop Proceedings*, pages 19–25. CEUR-WS.org.

Shashank Sharma, PYKL Srinivas, and Rakesh Chandra Balabantaray. 2015. Text normalization of code mix and sentiment analysis. In *2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015, Kochi, India, August 10-13, 2015*, pages 1468–1473.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA, October. Association for Computational Linguistics.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Gohneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of The First Workshop on Computational Approaches to Code Switching, held in conjunction with EMNLP 2014.*, pages 62–72, Doha, Qatar. ACL.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2):267–307, June.

Minh thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, pages 104–113.

Yogarshi Vyas, Spandana Gella, Jatin, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 974–979.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 90–94, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *CoRR*, abs/1502.01710.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A C-LSTM neural network for text classification. *CoRR*, abs/1511.08630.