# Deep Learning Techniques for Humor Detection in Hindi-English Code-Mixed Tweets

**Sushmitha Reddy Sane*[1] Suraj Tripathi*[2] Koushik Reddy Sane[1] Radhika Mamidi[1]**
[1]International Institute of Information Technology, Hyderabad
[2]Indian Institute of Technology, Delhi
{sushmithareddy.sane, koushikreddy.sane}@research.iiit.ac.in,
surajtripathi93@gmail.com, radhika.mamidi@iiit.ac.in

## Abstract

We propose bilingual word embeddings based on word2vec and fastText models (CBOW and Skip-gram) to address the problem of Humor detection in Hindi-English code-mixed tweets in combination with deep learning architectures. We focus on deep learning approaches which are not widely used on code-mixed data and analyzed their performance by experimenting with three different neural network models. We propose convolution neural network (CNN) and bidirectional long-short term memory (biLSTM) (with and without Attention) models which take the generated bilingual embeddings as input. We make use of Twitter data to create bilingual word embeddings. All our proposed architectures outperform the state-of-the-art results, and Attention-based bidirectional LSTM model achieved an accuracy of 73.6% which is an increment of more than 4% compared to the current state-of-the-art results.

## 1 Introduction

In the present day, we observe an exponential rise in the number of individuals using Internet Technology for different purposes like entertainment, learning and sharing their experiences. This led to a tremendous increase in content generated by users on social networking and micro-blogging sites. Websites like Facebook, Twitter, and Reddit (Danet and Herring, 2007) act as a platform for users to reach large masses in real-time and express their thoughts freely and sometimes anonymously amongst communities and virtual networks. These natural language texts depict various linguistic elements such as aggression, irony, humor, and sarcasm. In recent years, automatic detection of these elements (Davidov et al., 2010) has become a research interest for both organizations and research communities.

The advancement in computer technologies places increasing emphasis on systems and models that can effectively handle natural human language. So far, the majority of the research in natural language processing and deep learning is focused on the English language as individuals across the world use it widely. But, in multilingual geographies like India, it is a natural phenomenon for individuals to use more than one language words in speech and in social media sites like Facebook and Twitter (Cárdenas-Claros and Isharyanti, 2009; Crystal, 2002). Data shows that in India, there are about 314.9 million bilingual speakers and most of these speakers tend to mix two languages interchangeably in their communication. Researchers (Myers-Scotton, 1997) defined this linguistic behavior as Code-mixing - the embedding of linguistic units such as phrases, words, and morphemes of one language into an utterance of another language which produces utterances consisting of words taken from the lexicons of different languages.

The primary challenge with the code-mixed corpus is the lack of data in general text-corpora, (Nguyen and Doğruöz, 2013; Solorio and Liu, 2008a,b) for conducting experiments. In this paper, we take up the task of detecting one critical element of natural language (Kruger, 1996) which plays a significant part in our linguistic, cognitive, and social lives, i.e., Humor. Martin (Martin and Ford, 2018) extensively studied the psychology of humor and stated that it is ubiquitous across cultures and it is a necessary part of all verbal communication. The classification of some text as humor can be very subjective. Also, capturing Humor in higher order structures (de Oliveira et al., 2017) through text processing is considered as a challenging natural language problem. Pun detection in one-liners (Kao et al., 2016) and detection of humor in Yelp reviews (de Oliveira et al., 2017)

* These authors contributed equally to this work.

https://en.wikipedia.org/wiki/Multilingualism_in_India



*Proceedings of the 10th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 57–61

have also been studied in recent years.

Deep learning techniques (LeCun et al., 2015) have contributed to significant progress in various areas of research, including natural language understanding. Convolutional neural network based networks have been used for sentence classification (Kim, 2014), bidirectional LSTM networks (biLSTM) were used for sequence tagging (Huang et al., 2015), and attention based bidirectional LSTM networks were used for relational classification (Zhou et al., 2016) and topic-based sentiment analysis (Baziotis et al., 2017). In this work, we propose three deep learning networks using bilingual word embeddings as input and compare it against the classification models presented in (Khandelwal et al., 2018) using their annotated corpus to detect one of the playful domains of language: Humor. An example from the corpus:

"*Subha ka bhula agar sham ko wapas ghar aa jaye then we must thank GPS technology.*."
"(If someone is lost in the morning and returns home in the evening then we must thank GPS technology.)

This tweet is annotated as humorous. In particular, we are focused on code-mixed data as it lacks the presence of bilingual word embeddings, commonly used, to train any deep learning model which is essential for understanding human behavior, events, reviews, studying trends as well as linguistic analysis (Vyas et al., 2014).

### 1.1 Corpus creation for Bilingual Word Embeddings

The corpus used for training the bilingual word embeddings is created using Twitter's API. Around 200k tweets are extracted using 1000 most common words from the training corpus after removing stop words. Preprocessing is done on the sentences, and Twitter handles starting with "@" or words that have any special symbol are removed. URLs are replaced with the word "URL".

### 1.2 Word2Vec

Code-mixed (Hindi-English) data need vector representations of its words to train a deep learning based model. However, our corpus being bilingual in nature prohibits the use of any pre-trained word2vec (Mikolov et al., 2013) representations. As mentioned earlier, we used the collected Twitter data to train the bilingual word embedding model. We experimented with various hyperparameters like embedding size, window length, and negative sampling. Based on the results, we finalized the following set of values for our main task - humor detection.

- Embedding size: 300, Window length: 10, Negative sampling

### 1.3 FastText

One of the limitations of word2vec model is the inability to handle words with very low frequency in the training corpus and out-of-vocabulary words which might be present in the unseen text instances. Example: people on social media write words like "happppyyyyy", "lolll", etc. These kinds of new words can't have pre-trained word embeddings. To address this problem in the bilingual scenario, we analyzed the performance of fastText (Bojanowski et al., 2017) word embedding model, which considers subword information, for generating word embeddings. FastText learns character n-gram (Joulin et al., 2016) representations and represents words as the sum of the n-gram vectors, where n is a hyperparameter. We kept hyperparameters like embedding size, window length, etc., same as in word2vec model to compare their results.

### 1.4 Model Architecture

We propose three different deep learning architectures for the task of humor detection based on CNN and biLSTM networks which take bilingual word embeddings as input. We used cross-entropy loss function and Adam optimizer for training all our proposed architectures.

#### 1.4.1 Model 1 - Convolutional Neural Network (CNN)

We propose a CNN-based model, refer to Figure 1, which takes bilingual word embeddings as input. CNN based model makes use of a set of 4 parallel 1D convolution layers to extract features from the input embeddings. Features derived from the convolution layer are then fed into the global max-pool layer, which extracts one feature per filter. The extracted features from max-pool layer are then flattened and passed to multiple fully connected (FC) layers. Finally, classification is performed using a softmax layer. We have used various training techniques such as dropout (.25 to .75) (Srivastava et al., 2014) and batch-normalization (Ioffe and Szegedy, 2015) that helps

in reducing overfitting and sensitivity towards initial weights respectively. With the use of batch-normalization, we also observed improvement in the convergence rate.

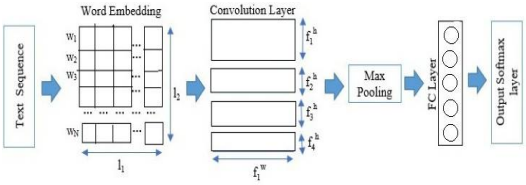


Figure 1: Proposed CNN architecture

The model uses four parallel instances of convolution layer with varying kernel sizes. We experimented with various values for hyperparameters such as the number of kernels, kernel sizes and finalized following values based on the performance on the validation set:

- Kernel size:

$$f_1^h = 3, f_2^h = 6, f_3^h = 9, f_4^h = 12$$

- Number of kernels = 200, Stride = 1.

We analyzed the performance of the proposed CNN based network with both word2vec and fastText generated bilingual word embeddings and presented their results in Table 2. Here, Model 1(a) refers to CNN with word2vec and Model 1(b) refers to CNN with fastText based word representations respectively.

#### 1.4.2 Model 2 - Bidirectional LSTM Network

Bidirectional LSTM architectures have been proved to be very useful to model word sequences and are robust to learn on data with long-range temporal dependencies. We use the bidirectional LSTM network on the input bilingual word embeddings to capture the compositional semantics for the bilingual texts in our experiments.

The sentiment of each word in the sentence depends on the context in which the word is used, where context includes content in front of the word as well as behind the word. To model these scenarios, we make use of bidirectional LSTM network which has been successfully applied in generating context-dependent hidden representations as well as capturing long-term dependencies in text classification tasks (Wang et al., 2016). We experimented with a different number of hidden layers and number of hidden units in each hidden layer and finalized the value of 1 and 200 respectively based on the results on the validation set. We used similar architecture as showed in Figure 2 with no attention mechanism and used concatenated $\overrightarrow{h_t}$ and $\overleftarrow{h_1}$ as input to the dense layer (#hidden units = 200) which is followed by the final softmax layer. To analyze the effect of different bilingual word embeddings, we make use of both word2vec, and fastText generated embeddings. In Table 2, Model 2(a) refers to the use of word2vec with Model 2 and Model 2(b) refers to the use of fastText with Model 2.

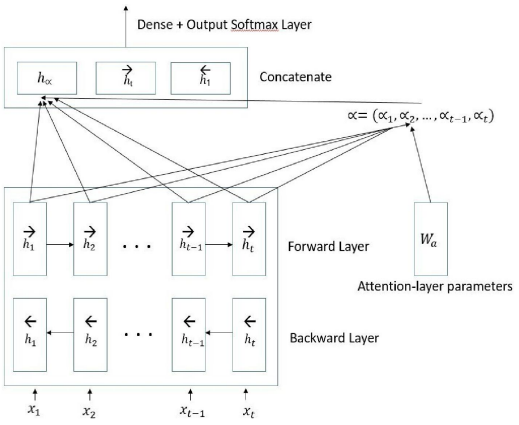


Figure 2: Proposed Attention based BiLSTM Model

#### 1.4.3 Model 3 - Attention-based Bidirectional LSTM Network

We further propose an attention-based mechanism for the bidirectional LSTM network. The word-level attention model learns which words in a given sentence are more critical for determining the overall emotion (humorous / non-humorous) of the sentence. These words act as decisive points. Some parts in sentences create noise, and this mechanism helps to filter out those noises.

Input sentence $x_1, x_2, ..., x_{t-1}, x_t$ represents bilingual word embedding of the input text utterance, which is fed into the hidden layer of the proposed bidirectional LSTM network as input. As presented in Figure 2, bidirectional LSTM architecture makes use of both forward and backward hidden states at each time step. We used well-known standard LSTM units for our architecture and thus omitted the equations related to the cell units. At each time step $i$, $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$ represents complete hidden state representation. We make use of the hidden state of each step to calculate the weights for each word and weighted summation of all time steps. $h_\alpha$ is used as an input to the

classifier in combination with $\overrightarrow{h_t}$ and $\overleftarrow{h_1}$. Concatenated hidden states are passed on to a single dense layer, followed by output softmax layer. We used the same hyperparameter settings for hidden representation and dense layer size as mentioned in Model 2 to analyze the effect of adding attention to the proposed model. We experimented with both word2vec and fastText generated bilingual word embeddings, and results are presented in Table 2. Here, Model 3(a) refers to the use of word2vec and Model 3(b) refers to the use of fastText with Model 3 respectively.

| Word-Hi | Word-En | Word2Vec | FastText |
|---|---|---|---|
| pyaar | love | 0.64 | 0.78 |
| nafrat | hate | 0.71 | 0.73 |
| ldai | fight | 0.74 | 0.85 |
| majak | funny | 0.62 | 0.71 |
| gussa | angry | 0.78 | 0.76 |

Table 1: Similar meaning Hindi and English words similarity scores with word2vec and fastText models

## 2 Results

The benchmark dataset that is published online by (Khandelwal et al., 2018) is used for evaluating the effectiveness of bilingual word embeddings and proposed deep learning models. It contains 3543 annotated tweets where 1755 are labeled humorous and 1698 as non-humorous. We make use of 5-fold cross-validation for generating our experimental results. Using all the features (Khandelwal et al., 2018), the baseline systems: kernel SVM, random forest, extra tree, and naive Bayes presented the best accuracy of 69.3%. Going forward, to the best of our knowledge, we are the first to experiment with deep learning architectures using bilingual word embedding for detecting humor in code-mixed data. All of our models showed better accuracies than current state-of-art-results, and our proposed Attention-based bidirectional LSTM achieved the best accuracy of 73.6%.

The challenges in this task are the linguistic complexity of code-mixed data and lack of clean data. To address phrasal repetitions, short and simple constructions, non-grammatical words and spelling errors in the data, larger corpora need to be built and annotated in the geographies and communities where multilingualism is observed.

In Table 1, we analyzed the generated bilingual word embeddings by comparing the similarity scores of Hindi and English words with similar meaning. We observed that fastText model showed better similarity scores than word2vec model which indicates that bilingual word embeddings do get better with subword information which is used in learning fastText word representations. In Table 2, we presented the results of our proposed deep learning based architectures which takes bilingual words embeddings generated from word2vec and fastText skip-gram model. We also experimented with CBOW versions of both learning strategies and achieved similar results.

| Model | Accuracy |
|---|---|
| Random Forest* | 65.2 |
| Naive Bayes* | 67.2 |
| Extra tree* | 67.8 |
| Kernel SVM* | 69.3 |
| Model 1(a) | 70.8 |
| Model 1(b) | 71.3 |
| Model 2(a) | 71.5 |
| Model 2(b) | 72.2 |
| Model 3(a) | **72.8** |
| Model 3(b) | **73.6** |

Table 2: Detailed accuracies achieved on the benchmark dataset by different models. *Random Forest, Naive Bayes, Extra tree, and kernel SVM accuracies are from (Khandelwal et al., 2018)

## 3 Conclusion

In this paper, we address the problem of humor detection in code-mixed Hindi-English data generated by bilingual users. We propose three different deep learning based models which take bilingual word embeddings as input. Both word2vec and fastText based models are used for learning bilingual word representations and also to demonstrate the effectiveness of these techniques by presenting similarity scores of words with similar meaning in Hindi and English languages. The proposed attention-based biLSTM model worked best with an accuracy of 73.6%. Compared to the state-of-the-art models all our proposed deep learning models performed better at detecting humor in code-mixed data. For future work, we will generate aligned multilingual word embeddings and compare them with vectors aligned with MUSE, and pre-aligned fastText embeddings.

https://github.com/facebookresearch/MUSE

## References


Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Mónica Stella Cárdenas-Claros and Neny Isharyanti. 2009. Code-switching and code-mixing in internet chatting: Betweenyes, ya,andsi-a case study. *The Jalt Call Journal*, 5(3):67–78.

David Crystal. 2002. Language and the internet. *IEEE Transactions on Professional Communication*, 45(2):142–144.

Brenda Danet and Susan C Herring. 2007. *The multilingual Internet: Language, culture, and communication online*. Oxford University Press on Demand.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Justine T Kao, Roger Levy, and Noah D Goodman. 2016. A computational model of linguistic humor in puns. *Cognitive science*, 40(5):1270–1285.

Ankush Khandelwal, Sahil Swami, Syed S Akhtar, and Manish Shrivastava. 2018. Humor detection in english-hindi code-mixed social media content: Corpus and baseline system. *arXiv preprint arXiv:1806.05513*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Arnold Kruger. 1996. The nature of humor in human nature: Cross-cultural commonalities. *Counselling Psychology Quarterly*, 9(3):235–241.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436.

Rod A Martin and Thomas Ford. 2018. *The psychology of humor: An integrative approach*. Academic press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Dong Nguyen and A Seza Doğruöz. 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.

Luke de Oliveira, ICME Stanford, and Alfredo Láinez Rodrigo. 2017. Humor detection in yelp reviews.

Thamar Solorio and Yang Liu. 2008a. Learning to predict code-switching points. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 973–981. Association for Computational Linguistics.

Thamar Solorio and Yang Liu. 2008b. Part-of-speech tagging for english-spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1051–1060. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, 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)*, pages 974–979.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212.