

# Humor Generation and Detection in Code-Mixed Hindi-English

**Kaustubh Agarwal, Rhythm Narula**

Department of Computer Science and Engineering

Netaji Subhas University of Technology

New Delhi, India

{kaustubh, rhythm.narula}.co19@nsut.ac.in

## Abstract

Computational humor generation is one of the hardest tasks in natural language generation, especially in code-mixed languages. Existing research has shown that humor generation in English is a promising avenue. However, studies have shown that bilingual speakers often appreciate humor more in code-mixed languages with unexpected transitions and clever word play. In this study, we propose several methods for generating and detecting humor in code-mixed Hindi-English. Of the experimented approaches, an Attention Based Bi-Directional LSTM with converting parts of text on a word2vec embedding gives the best results by generating 74.8% good jokes and IndicBERT used for detecting humor in code-mixed Hindi-English outperforms other humor detection methods with an accuracy of 96.98%.

## 1 Introduction

Humor is one of the most powerful tools of communication capable of transforming negative sentiments and emotions to positive ones (Savage et al., 2017, Castro et al., 2016). From the early 1990's, computer science researchers and linguistic scholars have attempted to generate humor using computers giving rise to the field of computational humor generation. However, the task of humor generation has proven to be one of the most difficult tasks in natural language generation due to the lack of fundamental knowledge of how brains perceive and generate humor and its intimate link with the inner workings of the brain (Binsted et al., 2006).

While the field of computational humor generation is nowhere near saturation, deep learning techniques have shown promising results (Ren and Yang, 2017). Meanwhile, studies have shown that appreciating humor in jokes involves resolution and incongruity-detection, both of which require a strong command over the medium of communication of the joke (Ayçiçeği-Dinn et al., 2018).

The added element of surprise and clever word-play by rhyming a logically apt word from another language in a joke adds to its humor (Schröter, 2005). These studies and observations show that code-mixed joke generation can prove effective in creating humor. Code-mixing refers to the insertion of words, phrases, and morphemes of one language into a statement or an expression of another language (Alex, 2008).

India has around 350 million people that speak code-mixed Hindi-English (popularly known as Hinglish) (GTS, 2019), and according to David Crystal, a British linguist at the University of Wales, the world's Hinglish speakers may soon outnumber native English speakers. This makes Hinglish one of the most widely spoken code-mixed languages worldwide.

In this paper, we aim to generate and detect humor in code-mixed Hindi-English. Although a lot of work has been carried out on humor generation and detection in English (Zhang et al., 2020, Petrović and Matthews, 2013, Winters et al., 2019, Dybala et al., 2010, Winters et al., 2018), and humor detection in code-mixed Hindi-English (Khandelwal et al., 2018), humor generation in code-mixed languages is relatively unexplored. To the best of our knowledge, we are the first to attempt humor generation in code-mixed languages.

The contributions of our work include

1. Exploration of novel methodologies to generate humor in code-mixed Hindi-English.
2. Using transformers and other deep learning models to create classifiers to detect humor in code-mixed Hindi-English.

## 2 Related Works

Early papers date back to 1994 when Graeme Ritchie and Kim Binsted designed a computer

program called JAPE (Joke Analysis and Production Engine) which was used to generate question-answer type of puns. It was in 2005 when Graeme Ritchie introduced various pun generation mechanisms. They divided puns into two major categories of self-contained puns which are portions of text that can be used as humorous objects in a wide variety of situations. Their best preconditions to be used are preferred know-how (of the surrounding lifestyle, and so on) and contextually integrated puns which are used with a broader discourse and some more text (Ritchie, 2005).

Past endeavors using template-based approaches depend on formats and several constraints which intensely compel the quality and variety of generated sentences. Most past research is focused towards using layouts which are helpful for generating template based jokes, but they are often rigid. None of the frameworks intended to be innovative as the skeletons of the sentences are fixed. Owing to this, over time the quest of joke generation has shifted from template-based approaches to neural networks.

Various work has been done for pun generation using the LSTM approach. The initial model was proposed by Ren and Yang (2017)'s current affair jokes generator which was trained on a joke corpus along with news data for current affairs using LSTM RNN as architecture and GloVe as vector representation for training data (Amin and Burghardt, 2020). The results however, were not satisfactory. Another such attempt was made by using a conditioned neural language model that was trained using general text corpus which was then passed through a decoding algorithm (Yu et al., 2018). Though these neural network techniques gave creative outputs, they lacked humor in comparison to template-based approaches which showed fairly good humor despite lacking creativity. In this paper we take a deeper dive into humor generation using neural networks and attempt to generate humor using deep learning models in code-mixed Hindi-English.

### 3 Proposed Methodology

The proposed methodology consists of two sections, each covering a major contribution of this paper.

#### 3.1 Humor Detection

For the purpose of humor detection, deep learning models and transformers have been used. To create embeddings, we have scraped a dataset of 412k Hinglish tweets from Twitter and pre-processed them by removing rare words, hashtags, mentions and Uniform Resource Locators (URLs) to create word embeddings. For better comparison of results, we have used the dataset of 3453 jokes in code-mixed Hindi-English created by Khandelwal et al. (2018).

**mBERT** BERT is a bi-directionally trained language model which helps it to obtain a deeper sense of the language flow compared to single direction trained models (Devlin et al., 2019). mBERT takes a multilingual dataset to train the model. It uses a multi-layer bidirectional transformer encoder which is capable of learning deep bidirectional representations and can easily be fine-tuned for several tasks including humor detection. All the more explicitly, given a sentence, our approach first obtains its token representation which is fed to the output layer which acts as a binary classifier by using class labels. The final output layer generates scores which are used to further train the model by calculating the mean squared error between the predicted score and label (Mao and Liu, 2019).

**IndicBERT** Just like mBERT, IndicBERT is also used to solve a variety of NLP tasks in the multilingual context. IndicBERT is a multilingual ALBERT model pre-trained solely on 12 significant Indian dialects consisting of about 9 billion tokens and evaluated using a variety of tasks. It has much fewer parameters as compared to other multilingual models like mBERT (Kakwani et al., 2020) and performs better than mBERT on most tasks involving Indic languages. The major advantage of this model is that it performs well even with small training datasets and excels in doing cross-lingual learning which is beneficial while using Indian languages which lack these resources.

#### 3.2 Humor Generation

We have explored humor generation in code-mixed Hindi-English using two methods.

##### 3.2.1 Training deep learning models on Hindi-English jokes

The most direct method of generating humor in code-mixed Hindi-English is by using deep learn-

ing models to generate text on a dataset of code-mixed Hindi-English jokes. However, training deep learning models on Hindi-English jokes by using English word embeddings leads to unintelligible words and jokes due to lack of understanding of Hindi words. To resolve this problem, we have used the embeddings from the 412K Hinglish tweets mentioned earlier. We have used Word2Vec (Mikolov et al. (2013a), Mikolov et al. (2013b)) and FastText (Bojanowski et al., 2017, Joulin et al., 2017) for generating word embeddings.

### 3.2.2 Converting English jokes to code-mixed Hindi-English

For this method, we have used deep learning models trained on a dataset of 200K English jokes created by Pungas (2017). The embeddings are created using Word2Vec and FastText but only for English text instead of Hindi-English text. The generated jokes are purely in English. We have experimented with two methods to convert this English text (jokes) to code-mixed Hindi-English:

1. Converting parts of text: Using a part-of-speech tagger, singular and plural nouns have been converted from each English joke to Hindi. While converting multiple other parts-of-speech have been tested, it often leads to unintelligible results in the context of the entire sentence.
2. Using Pre-trained Encoder and Transfer Learning: A Semi-supervised Approach to Generate Code-Mixed Text using Pre-trained Encoder and Transfer Learning as proposed by Gupta et al. (2020) is used to convert jokes generated in English to code-mixed Hindi-English jokes.

The jokes generated by these techniques contain code-mixed elements of Hindi and English.

The following architectures have been used to create models for each of the aforementioned methods

1. LSTM: Hochreiter and Schmidhuber (1997)
2. Bi-directional LSTM: Schuster and Paliwal (1997)
3. Attention Based Bi-directional LSTM: Zhou et al. (2016)

For generating jokes using these models, the first word or sentence could be provided as an input or could be chosen randomly by the model. While choosing a word would be ideal for generating jokes on a particular topic, randomly choosing words would give room for creativity. Ideally, the model should then pick the word with highest probability of occurring in a given context since that would lead to a better joke based on its learning from the training data. However, this could lead to boring and conservative predictions (Ren and Yang, 2017). This can be countered by using weighted-pick strategies that encourages the model to take greater risks at the cost of getting poor jokes more often. The model thus picks random words from the vocabulary according to the weights, which are the probabilities coming from the last softmax layer.

## 4 Experimental Results

### 4.1 Humor Detection

The dataset of jokes in code-mixed Hindi-English created by Khandelwal et al. (2018) is used for training transformers and deep learning models for detecting humor. Each transformer is trained for a total of 5 epochs, and each deep learning model has been trained for 20 epochs with a 15 percent validation split for each.

The hyper-parameters used for all RNNs are: dropout for recurrent state = 0.2, dropout for input state = 0.2, number of LSTM units = 150

Each transformer and deep learning model performs the task of binary classification to detect whether a particular text is humorous or not. Deep learning models have been given the word embeddings of Hindi-English text as described earlier. As evident from Table-1, IndicBERT gives the best classification results among the tested architectures, and transformers significantly outperformed other deep learning models.

Model	Accuracy
LSTM	74.48%
Bi-Directional LSTM	74.80%
Attention Bi-Directional LSTM	79.31%
mBERT	95.37%
IndicBERT	96.98%

Table 1: Humor Detection Accuracies on the dataset of jokes in code-mixed Hindi-English created by Khandelwal et al. (2018)

Model	Percentage of Good Jokes	
	Word2Vec	FastText
LSTM	68.8%	67.2%
Bi-Directional LSTM	69%	68.9%
Attention Based Bi-Directional LSTM	72.4%	71.3%

Table 2: Percentage of good jokes generated by training deep learning models on Hindi-English jokes

Thus, with a validation accuracy of 96.98%, IndicBERT outperforms mBERT that has a validation accuracy of 95.37%.

## 4.2 Humor Generation

We have experimented with two different types of humor generation techniques. In both techniques, the hyper-parameters for the RNNs are: dropout for recurrent state = 0.2, dropout for input state = 0.2, number of LSTM units = 300.

### 4.2.1 Training deep learning models on Hindi-English jokes

For creating jokes using a dataset of Hindi-English jokes, we experimented with Word2Vec and FastText embeddings created using a dataset of 412k Hinglish tweets from Twitter. We generated 1000 jokes using each deep learning model and ran it on the IndicBERT classifier created for humor detection to find the percentage of good jokes generated by the deep learning model. Table-2 shows the percentage of good jokes created using this joke generation method for each deep learning model.

Among the tested architectures, Attention Based Bi-Directional LSTM gives the best result with 72.4% good jokes with Word2Vec embeddings. It is also observed that Word2Vec gives slightly better results than the corresponding FastText embeddings. Possible reasons for this observation may include the fact that code-mixed data does not allow n-grams to be the primary criterion for classifica-

tion.

### 4.2.2 Converting English jokes to code-mixed Hindi-English

For creating jokes in English and converting them to code-mixed Hindi-English we again experimented with Word2Vec and FastText embeddings but only for English. From each deep learning model, 1000 jokes are generated. These 1000 jokes in English are converted to code-mixed Hindi-English by two methods- converting parts of text, and using Pre-trained Encoder and Transfer Learning. The obtained code-mixed Hindi-English jokes are passed through the previously created IndicBERT classifier. The percentage of good jokes generated are shown in Table-3.

It can be observed from Table 3 that Attention Based Bi-directional LSTM created from English jokes converted to code-mixed Hindi-English by converting parts of text gives the best results by creating 74.8% good jokes. It can be deduced that usage of unexpected and unusual words adds to the element of surprise in between an English joke and adds to the element of humor more than converting the entire sentence to a logical code-mixed joke.

Table-4 gives samples of good and bad jokes generated from each of the methods. All Hindi words are written in the Devanagari script to differentiate between English and Hindi words used in a joke. It is noteworthy that a substantial amount of humor is lost in translation of the joke to English

Model	Percentage of Good Jokes			
	Converting Parts Of Text		Using Pre-trained Encoder and Transfer Learning	
	Word2Vec	FastText	Word2Vec	FastText
LSTM	69.4%	69.1%	69.2%	68.8%
Bi-directional LSTM	71.5%	70.9%	70.7%	70.2%
Attention Based Bi-directional LSTM	74.8%	73.2%	74.2%	73.6%

Table 3: Percentage of good jokes generated by converting English jokes to code-mixed Hindi-English

Method	Good Joke	Translation	Bad Joke	Translation
Training deep learning models on Hindi-English jokes	Aunty जो सोचती हैं उसकी परवाह क्यों करें... troll किया तो डरने क्या	Why should we care what aunty thinks? Why worry if we trolled?	कितने भी phone बनाते हैं वे एक जैसे दिखेंगे	How many ever phones they make all will look the same.
Converting English jokes to code-mixed Hindi-English by converting parts of text	I call my गणित संकट	I call my mathematics trouble	What's a राक्षस 's favorite खाना? डाक का कबूतर	What's a monster's favorite food? The messenger pigeon
Converting English jokes to code-mixed Hindi-English using pre-trained encoder and transfer learning	क्या आप bear ki हत्या करना चाहते हैं? Personally, I would पर I can bear-ly see it.	Do you want to murder a bear? Personally, I would but I can bear-ly see it.	आप बस attention चाहते हैं. Teenage girl है क्या	You just want attention. Are you a teenage girl?

Table 4: Samples of good and bad jokes generated by each method in native scripts

as presented in the table.

## 5 Conclusion and Future Work

In this paper, we propose novel approaches for humor generation in code-mixed Hindi-English and explored the idea of using deep learning models for generating and detecting humor in it. One of the proposed approaches uses LSTM, Bi-Directional LSTM and Attention Based LSTM for generating jokes in code-mixed Hindi-English using Hinglish word embeddings and the other uses the same deep learning approaches to generate jokes in English which are further transformed into code-mixed Hindi-English by converting selective parts of text or using a pre-trained encoder and transfer learning method. The produced jokes were evaluated using IndicBERT which performed better than other humor detection models with an accuracy of 96.98%. The proposed conversion of parts of a joke generated in English gave the best results in humor generation by producing 74.8% good jokes on word2vec embedding. Compared to the state of the art humor detection models, our proposed approach performed better at detecting humor on code-mixed Hindi-English data. Since humor generation in code-mixed Hindi-English is unexplored, this paper proposes a baseline for future work. Future work may include extension of humor generation to other code-mixed languages, usage of contextualized embeddings, and better deep learning techniques that understand humor more deeply. Furthermore, better models can be built using iterative data augmentation in such low-resource sce-

narios. Appropriate usage of phonetic models to generate similar sounding words with contrasting meanings in the parent code-mixed languages under consideration could be another potential avenue for future research.

## References

- Beatrice Alex. 2008. *Automatic Detection of English Inclusions in Mixed-lingual Data with an Application to Parsing*. Ph.D. thesis.
- Miriam Amin and Manuel Burghardt. 2020. [A survey on approaches to computational humor generation](#). In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 29–41, Online. International Committee on Computational Linguistics.
- Ayşe Ayçiçeği-Dinn, Simge Şişman-Bal, and Catherine L Caldwell-Harris. 2018. [Are jokes funnier in one's native language?](#) *HUMOR*, 31(1):5–37.
- K. Binsted, B. Bergen, S. Coulson, A. Nijholt, O. Stock, C. Strapparava, G. Ritchie, R. Manurung, H. Pain, A. Waller, and D. O'Mara. 2006. [Computational humor](#). *IEEE Intelligent Systems*, 21(2):59–69.
- 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.
- Santiago Castro, Matías Cubero, Diego Garat, and Guillermo Moncecchi. 2016. [Is this a joke? detecting humor in spanish tweets](#). In *Lecture Notes in Computer Science*, pages 139–150. Springer International Publishing.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Pawel Dybala, Michal Ptaszynski, Jacek Maciejewski, Mizuki Takahashi, Rafal Rzepka, and Kenji Araki. 2010. [Multiagent system for joke generation: Humor and emotions combined in human-agent conversation](#). *Journal of Ambient Intelligence and Smart Environments*, 2:31.
- GTS. 2019. [Hinglish - the Biggest Language you've Never Heard of with 350 Million Speakers](#).
- Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. [A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- 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.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. [IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.
- A. Khandelwal, Sahil Swami, S. Akhtar, and Manish Shrivastava. 2018. [Humor detection in english-hindi code-mixed social media content : Corpus and baseline system](#). *ArXiv*, abs/1806.05513.
- Jihang Mao and W. Liu. 2019. [A bert-based approach for automatic humor detection and scoring](#). In *Iber-LEF@SEPLN*.
- Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013a. [Efficient estimation of word representations in vector space](#).
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. [Distributed representations of words and phrases and their compositionality](#). In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.
- Saša Petrović and David Matthews. 2013. [Unsupervised joke generation from big data](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–232, Sofia, Bulgaria. Association for Computational Linguistics.
- Taivo Pungas. 2017. [A dataset of english plaintext jokes](#).
- H. Ren and Quan Yang. 2017. [Neural joke generation](#).
- Graeme Ritchie. 2005. [Computational mechanisms for pun generation](#). In *Proceedings of the Tenth European Workshop on Natural Language Generation (ENLG-05)*.
- Brandon M. Savage, Heidi L. Lujan, Raghavendar R. Thipparthi, and Stephen E. DiCarlo. 2017. [Humor, laughter, learning, and health! a brief review](#). *Advances in Physiology Education*, 41(3):341–347.
- Thorsten Schröter. 2005. [Shun the pun, rescue the rhyme? : The dubbing and subtitling of language play in film](#).
- M. Schuster and K.K. Paliwal. 1997. [Bidirectional recurrent neural networks](#). *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- Thomas Winters, Vincent Nys, and Daniel Schreye. 2018. [Automatic Joke Generation: Learning Humor from Examples](#), pages 360–377.
- Thomas Winters, Vincent Nys, and Danny De Schreye. 2019. [Towards a general framework for humor generation from rated examples](#). In *Proceedings of the Tenth International Conference on Computational Creativity, ICCO 2019, Charlotte, North Carolina, USA, June 17-21, 2019*, pages 274–281. Association for Computational Creativity (ACC).
- Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. 2018. [A neural approach to pun generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1660, Melbourne, Australia. Association for Computational Linguistics.
- Hang Zhang, Dayiheng Liu, Jiancheng Lv, and Cheng Luo. 2020. [Let's be humorous: Knowledge enhanced humor generation](#). *ArXiv*, abs/2004.13317.
- 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)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.