How Effective is Incongruity? Implications for Code-mix Sarcasm Detection

Aditya Shah *
CSE department
IIT Indore
India
adishah3103@gmail.com

Chandresh Kumar Maurya
CSE department
IIT Indore
India
chandresh@iiti.ac.in

Abstract

The presence of sarcasm in conversational systems and social media like chatbots, Facebook, Twitter, etc. poses several challenges for downstream NLP tasks. This is attributed to the fact that intended meaning of a sarcastic text is contrary to what is expressed. Further, the use of code-mix language to express sarcasm is increasing day by day. Current NLP techniques for code-mix data have limited success due to the use of different lexicon, syntax, and scarcity of labeled corpora. To solve the joint problem of code-mixing and sarcasm detection, we propose the idea of capturing incongruity through sub-word level embeddings learned via fastText. Empirical results show that our proposed model achieves F1-score on code-mix Hinglish dataset comparable to pretrained multilingual models while training **10x** faster and using *lower memory footprint*.

1 Introduction

Sarcasm is defined as a sharp remark whose intended meaning is different from what it looks like. For example, "*I am not insulting you. I am describing you.*" could mean that the speaker is insulting the audience, but the receiver does not get it. Sarcasm usually involves ambivalence (also known as *incongruity* which means words/phrases having contradictory implications (Xiong et al., 2019) and difficult to comprehend. Though English is used as a way to communicate and exchange messages, majority of the people still use the mother language to express themselves on social media (Danet et al., 2007). According to one study (Hong et al., 2011), more than 50% posts on Twitter are written in a language other than English. *Code-switching* (also known as *code-mixing*) is a writing style in which the author uses words from different languages either in the same sentence (called *intra–sentential*) or different sentences (called *inter–sentential*) switching. An example of code-switching is: *"he said kal karte hai kaam""* (Gloss: he said tomorrow we'll do the work). The studies such as (Vizcaíno, 2011; Siegel, 1995) show that people use code-switch language when trying to convey comicality, satire or humor. Motivated by previous studies, we plan to detect sarcasm in code-mix languages. Though many studies exist to detect sarcasm in unimodal and multimodal data (Joshi et al., 2015, 2017; Carvalho et al., 2009; Xiong et al., 2019; Cai et al., 2019), methods to detect sarcasm in code-mix data are limited and have not been explored much. (Bansal et al., 2020; Aggarwal et al., 2020; Swami et al., 2018). It is due to several challenges such as *ambiguous words, variable lexical representation, word-level code-mixing, reduplication*, and *word-order*.

To solve some of these issues, we present a deep-learning based architecture to capture incongruity in code-mix data. Our proposed model achieves competitive performance as compared to pre-trained multilingual models (fine-tuned on the code-mix sarcasm detection task) with significantly *fewer parameters and faster training time*. Our contributions are as follows:

- Propose a deep learning based architecture along with sub-word level features to capture incongruity for sarcasm detection.
- Evaluate the performance of the proposed model on the Hindi-English (Hinglish) code-mix Twitter data that we collected. We further analyze existing multilingual models on the same. Our code+data will be available on [1].
- We will release the benchmark sarcasm

*Work done while the author was an intern at IIT Indore

[1] https://github.com/likemycode/codemix

Proceedings of the 18th International Conference on Natural Language Processing, pages 271–276
Silchar, India. December 16 - 19, 2021.

dataset for Hinglish language to facilitate further research on code-mix NLP.

2 Related Works

2.1 Learning Representation for Code-Mix Data

Models developed for multilingual representation learning have been explored for code-mix data representation by several authors (Winata et al., 2021; Khanuja et al., 2020; Aguilar et al., 2020a; Winata et al., 2018). Character-level representations have been utilized to address the out-of-vocabulary (OOV) issue in code-switch text (Winata et al., 2018), hand-crafted features were used in (Aguilar et al., 2019) for handling low-resource scenarios. Fine-tuning multilingual models like mBERT has shown to yield good results for various NLP tasks like Named-entity recognition (NER), part-of-speech (POS) tagging, etc., in (Khanuja et al., 2020), and surprisingly outperforms cross-lingual embeddings. Meta-embedding and hierarchical meta-embeddings have been found to be useful for closely-related language pairs in code-mix data (Winata et al., 2021) and usually outperform the mBERT (Khanuja et al., 2020). Char2Subword model proposed by (Aguilar et al., 2020b) builds representations from characters out of the subword vocabulary, and uses them to replace subwords in code-mix text (Winata et al., 2018), hand-crafted features were used in (Aguilar et al., 2019) for handling low-resource scenario. A centralized benchmark for Linguistic Code-switching Evaluation (LinCE) is released in (Aguilar et al., 2020a; Khanuja et al., 2020). Both of these works present results on several NLP tasks but *sarcasm detection*.

2.2 Sarcasm Detection in Code-Mix Data

There exists only a few works targeted towards sarcasm detection in code-mix data. In (Aggarwal et al., 2020), the author experiments with FastText (Joulin et al., 2016) and Word2Vec embeddings on two kinds of data: (1) Hinglish (Hindi-Eng) tweets, and (2) Hinglish+English tweets. They find that that Hinglish+English combination produces better results and achieves best F1 score of 79.4%. Various hand-crafted features, such as char n-grams, word n-grams etc., combined with random-forest/SVM are explored in (Swami et al., 2018) for sarcasm detection in code-mix, data and achieve F1-score of 78.4%. However, the dataset used is highly imbalanced with just 10% of sarcastic tweets and rest non-sarcastic. In such a scenario, the model might be biased towards predicting non-sarcastic tweets, and hence the evaluation results are quite skewed. Along similar lines, different switching features are used to form feature-vector and fed into a hierarchical attention network in (Bansal et al., 2020). They find that switching feature is a good indicator for irony/sarcasm/hate speech detection. However, none of these works handles *incongruity* explicitly or implicitly which has shown to achieve impressive results in sarcasm detection (Xiong et al., 2019).

3 Model Architecture

Code-mix language contains noisy words mixed with different languages and this might lead to out-of-vocabulary $< OOV >$ tokens. So, we use FastText skipgram (Bojanowski et al., 2016; Grave et al., 2018) for learning *subword level representation* from the code-mix data. We hypothesise that subword level representation is able to handle *ambiguous words*, *variable lexical representation*, and *word-level code-mixing*. For example, the ambiguous word “to” may be present in both Hindi and English language. Learning subword representation alleviates the problem of encoding the word “to” differently for both the languages. Further, variable length words like {“gharr”, “gharrr”, “gharrrr” will be split into tokens {“gha”,“har”, “arr”,“rrr” } and uniquely represented using only these subwords tokens. Code-mix words like “chapless” (Mix of Bengali “chap” and English “less”) are also represented via sub-words “chap” and “less”. The proposed model architecture is shown in Fig. 1.

Each sentence s is represented by its embedding $E = [e_1^T, e_2^T, \ldots, e_n^T]$, where $e_i \in R^d$ is the embedding vector and n is the length of the sentence. Inspired by the work of (Xiong et al., 2019), we propose the use of self-matching network in order to capture *incongruity* within the code-mix sentence. Specifically, for word-embedding pairs (e_i, e_j), we first calculate the joint feature vector $m_{i,j}$ via

$$m_{i,j} = GELU(e_i^T \cdot M_{i,j} \cdot e_j) \qquad (1)$$

where $M_{i,j} \in R^{d \times d}$ is the weight parameter matrix (learnable) and *GELU* (Hendrycks and Gimpel, 2016) is Gaussian Error Linear Unit activation function. Instead of *tanh* as used in (Xiong et al., 2019), we use *GELU* as it provides well-defined gradients in the negative region. Compared to RELU, since GELU is differentiable for all input values so it is

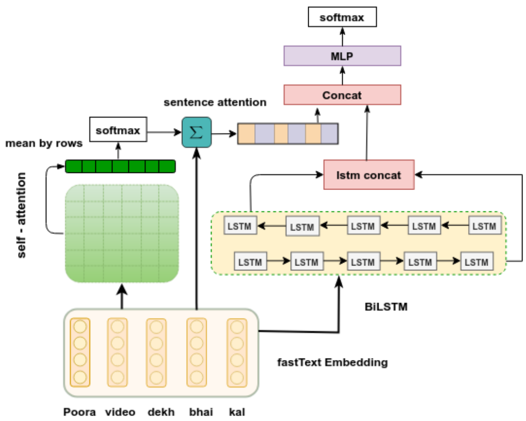


Figure 1: The model architecture

widely used in state-of-the-art NLP architectures. Our findings suggest that GELU activation yields better results for attention based mechanisms. Note that the above formulation is a form of *bilinear similarity* popularly used in *metric learning*. To calculate the attention score $\alpha_i, i \in (1, 2, \ldots, n)$ for each word, we take the mean of each row (contrary to *max* of rows as in (Xiong et al., 2019)) and apply the softmax for normalization.

$$\alpha_i = softmax(\mu(m_{1,i}), \mu(m_{2,i}), \ldots, \mu(m_{n,i})) \quad (2)$$

where $\mu()$ is the mean function, $m_{1,i}$ captures the incongruity of the first word with every other i^{th} word in the input text. Using the mean of all attention scores considers all the incongruous words present in the sentence for the computation. This helps the model to attend and learn from much larger span of incongruity. These attention scores denote how much weight should be assigned to each incongruous word. Next we calculate the weighted sentence attention vector v by:

$$v = \alpha^T E \quad (3)$$

Self-attention approach though captures the *incongruity* in the sentence, it misses the sentence's compositionality which is essential for sarcasm detection as suggested in (Tay et al., 2018). Therefore, sentence embedding is passed to the BiLSTM encoder (Graves et al., 2013) and the hidden states of the forward LSTM and backward LSTM are concatenated as

$$h_i = [\overrightarrow{LSTM}(e_i), \overleftarrow{LSTM}(e_i)], \forall i \in (1, 2, \ldots, n) \quad (4)$$

The output of the BiLSTM is concatenated with the sentence attention vector v and passed through the MLP layers with dropout. Finally, we pass it through softmax to predict the distribution over the binary labels (sarcasm vs non-sarcasm).

$$\hat{y} = softmax(MLP([v, h_n])) \quad (5)$$

where h_n is the hidden state corresponding to the last word in the forward and backward LSTM.

4 Experiments

4.1 The Dataset

The code-mix dataset used by (Aggarwal et al., 2020) is highly imbalanced with just 10% of sarcastic tweets and rest non-sarcastic. So, we create a dataset using TweetScraper built on top of scrapy [2] to extract code-mix hindi-english tweets. We pass search tags like #sarcasm, #humor, #bollywood, #cricket, etc., combined with most commonly used code-mix Hindi words as query. All the tweets with hashtags like #sarcasm, #sarcastic, #irony, #humor etc. are treated as positive. Non sarcastic tweets are extracted using general hashtags like #politics, #food, #movie, etc. The balanced dataset comprises of 166K tweets. We preprocess and clean the data by removing urls, hashtags, mentions, and punctuation in the data.

4.2 Baselines

The following baselines are used for comparison. (a) **Attention BiLSTM** (Aggarwal et al., 2020): The text features are extracted using word2vec and FastText which is fed to Series CNN, Parallel CNN, LSTM, Bi-LSTM and Attention Bi-LSTM, (b) **Multilingual Models**: To showcase the competitiveness of the proposed approach, we also compare with the state-of-the-art multilingual models like XLM-RoBERTa[3] and mBERT[4] from Huggingface library (Wolf et al., 2019). Specifically, we first fine-tune these models on the preprocessed code-mix corpus for mask language modeling task. Next, we use trained model by attaching a dense layer on top of it for detecting sarcasm in the code-mix tweets.

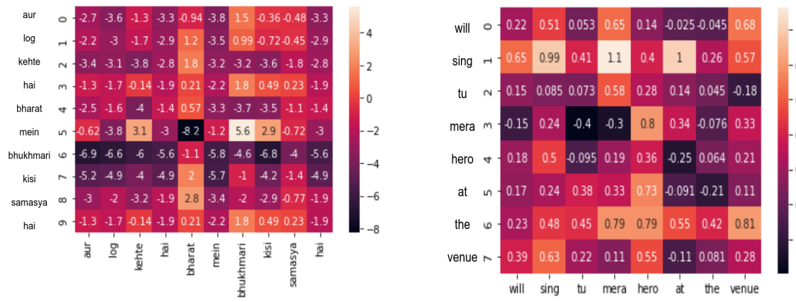
4.3 Experimental setup

For all the experiments, we use a train/valid/test split of 65:15:20. Categorical cross-entropy loss is minimized using adam optimizer for 15 epochs and learning rate of 5e-4 with step wise learning

[2]https://github.com/jonbakerfish/TweetScraper
[3]https://tinyurl.com/ydseww9d
[4]https://tinyurl.com/2dafn48n

Table 1: Comparative evaluation of the proposed approach.

Model	Recall	Prec.	Acc.	F1	Params	GPU	Train time
Attn. BiLSTM	77.34	81.24	80.21	79.34	21M	68 MB	0.8Hr
XLM-RoBERTa	86.17	91.48	89.04	88.75	278M	575 MB	8 Hr
mBERT	83.20	**94.55**	**89.17**	88.51	167M	483 MB	7 Hr
SelfNet (Ours)	**88.12**	88.25	89.04	**88.89**	35M	80 MB	1 Hr



(a) Sarcastic

(b) Non Sarcastic

Figure 2: Incongruity visualization via attention matrix

rate scheduler. FastText embedding size is 100 and the number of hidden units in BiLSTM and MLP layers are 256. We apply dropout of 0.4 along with gradient clipping of 0.3.

4.4 Results

The comparative evaluation results are shown in Table 1. We can see that the proposed approach achieves better F1 score than the baselines on the Hinglish code-mix data. In particular, our approach achieves around 10 points more F1 score than the Attn. BiLSTM of (Aggarwal et al., 2020). Additionally, it achieves slightly better F1 score than the pre-trained multilingual models.

Further, we also provide a comparison among all the 4 models in terms of the trainable parameters, GPU memory and training time in Table 1. As it can be seen that multilingual models require much larger memory and use almost 10x more parameters than our approach. Also, it is worth noting that the dataset size used to train our model is significantly less than the dataset size used to train multilingual models. Based on the comparison, we observe that our proposed model achieves a good balance between performance and model size for code-mix sarcasm detection. Figure 2 illustrates the output raw attention matrix P obtained before applying activation to visualize incongruous words.

As we can see from the Figure 2a, the words *bharat, mein, and bhukhmari* hold the highest incongruity (negative values). These 3 words define the semantics of the sentence and our model correctly attends to those words while finding the incongruity. Similarly for Figure 2b, there's no such incongruity present in the text. Thus the model does not assign high negative scores to this matrix.

4.5 Ablation Study

To evaluate the effectiveness of the network, we conduct an ablation study on the proposed architecture. This is summarized in Table 2. We test the self matching network proposed by (Xiong et al., 2019) which is referred to as Self Matching Net. Next, we replace biLSTM in our model with XLM-RoBERTa and mBert. The resulting models are denoted by "with XLM-RoBERTa" and "with mBert" respectively. The original Self Matching network does not perform so well on code-mix data as it only considers the most incongruous word pairs for prediction. Using mean operation helps to capture all the incongruous words which results in performance gain. Next when we replace BiLSTM with the multilingual models, the resulting approaches do not perform better than the proposed model. Although these models are trained on huge multilingual corpus, our study suggests that we can capture nuances of code-mix language using self-attention and simpler models like BiLSTM in a better way.

Table 2: Ablation Study

Models	Recall	Prec.	Acc.	F1
Self Matching Net	81.68	81.55	81.7	81.68
with XLM-RobertA	87.71	87.73	87.81	87.81
with mBERT	86.94	86.72	86.85	86.94
SelfNet (Ours)	**88.12**	**88.25**	**89.04**	**88.89**

5 Conclusion & Future work

In the present work, we propose the significance of incongruity in order to capture sarcasm in code-mix data. Our model effectively captures incongruity through FastText sub-word embeddings to detect sarcasm in the text. Empirical results on code-mix sarcasm data show that our approach performs satisfactorily compared to the multilingual models while saving memory footprint and training time. In future, we plan to work on a generalized model for other code-mix NLP tasks (NLI, NER, POS, QA etc) as well as test other code-mix languages like English - Spanish, English - Tamil, English - French etc.

References

Akshita Aggarwal, Anshul Wadhawan, Anshima Chaudhary, and Kavita Maurya. 2020. " did you really mean what you said?": Sarcasm detection in hindi-english code-mixed data using bilingual word embeddings. *arXiv preprint arXiv:2010.00310*.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020a. Lince: A centralized benchmark for linguistic code-switching evaluation. *arXiv preprint arXiv:2005.04322*.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. 2019. A multi-task approach for named entity recognition in social media data. *arXiv preprint arXiv:1906.04135*.

Gustavo Aguilar, Bryan McCann, Tong Niu, Nazneen Rajani, Nitish Keskar, and Thamar Solorio. 2020b. Char2subword: Extending the subword embedding space from pre-trained models using robust character compositionality. *arXiv preprint arXiv:2010.12730*.

Srijan Bansal, Vishal Garimella, Ayush Suhane, Jasabanta Patro, and Animesh Mukherjee. 2020. Code-switching patterns can be an effective route to improve performance of downstream nlp applications: A case study of humour, sarcasm and hate speech detection. *arXiv preprint arXiv:2005.02295*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515.

Paula Carvalho, Luís Sarmento, Mário J Silva, and Eugénio De Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's" so easy";-. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56.

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

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *ArXiv*, abs/1802.06893.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Lichan Hong, Gregorio Convertino, and Ed Chi. 2011. Language matters in twitter: A large scale study. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5):1–22.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762.

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

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. Gluecos: An evaluation benchmark for code-switched nlp. *arXiv preprint arXiv:2004.12376*.

Jeff Siegel. 1995. How to get a laugh in fijian: Code-switching and humor. *Language in Society*, pages 95–110.

Sahil Swami, Ankush Khandelwal, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A corpus of english-hindi code-mixed tweets for sarcasm detection. *arXiv preprint arXiv:1805.11869*.

Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856*.

María José García Vizcaíno. 2011. Humor in code-mixed airline advertising. *Pragmatics*, 21(1):145–170.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? *arXiv preprint arXiv:2103.13309*.

Genta Indra Winata, Chien-Sheng Wu, Andrea Madotto, and Pascale Fung. 2018. Bilingual character representation for efficiently addressing out-of-vocabulary words in code-switching named entity recognition. *arXiv preprint arXiv:1805.12061*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Tao Xiong, Peiran Zhang, Hongbo Zhu, and Yihui Yang. 2019. Sarcasm detection with self-matching networks and low-rank bilinear pooling. In *The World Wide Web Conference*, pages 2115–2124.