# A Corpus of English-Hindi Code-Mixed Tweets for Sarcasm Detection

**Sahil Swami**, **Ankush Khandelwal**, **Vinay Singh**, **Syed Sarfaraz Akhtar** and **Manish Shrivastava**

Language Technologies Research Centre, International Institute of Information Technology, Hyderabad

**Abstract.** Social media platforms like twitter and facebook have become two of the largest mediums used by people to express their views towards different topics. Generation of such large user data has made NLP tasks like sentiment analysis and opinion mining much more important. Using sarcasm in texts on social media has become a popular trend lately. Using sarcasm reverses the meaning and polarity of what is implied by the text which poses challenge for many NLP tasks. The task of sarcasm detection in text is gaining more and more importance for both commercial and security services. We present the first English-Hindi code-mixed dataset of tweets marked for presence of sarcasm and irony where each token is also annotated with a language tag. We present a baseline supervised classification system developed using the same dataset which achieves an average F-score of 78.4 after using random forest classifier and performing 10-fold cross validation.

## 1 Introduction

The Oxford dictionary[1] defines sarcasm as: "the use of irony to mock or convey contempt". Sarcasm generally has an implied negative statement but a positive surface sentiment [1]. As an example, consider the tweet: "*I'm so happy the teacher gave me all this homework right before Spring Break*". The author of this tweet uses positive words like 'happy' but it can be clearly observed that the author is not happy. Although sarcasm cannot be completely formally defined, it can be detected by humans in texts and speech. Sarcasm and irony,though different, are very closely related [2], so we consider them same in this paper. Twitter is one of the most used social media platforms used by people to express their opinion [3]. Many companies use this data for opinion mining and sentiment analysis to study the market. But a tweet may not always state the exact opinion of the user i.e. if it is sarcastically expressed. As it has become a common trend to use sarcasm on social media texts, detecting sarcasm in a tweet becomes more crucial for tasks like opinion mining and sentiment analysis.

Code-switching and code-mixing are two of the most commonly studied phenomena in multilingual societies [4]. Code-switching is generally inter-sentential

[1] http://www.oxforddictionaries.com/

while code-mixing is intra-sentential. Code-mixing refers to embedding linguistic units of one language into an utterance of another language. An example of a code-mixed sentence is: "*modi ji notebandi ki dikkat ko door karne k liye 200 rupay ka note bi market me laao. Chae 50 ka band ho jaye*". Words such as 'market' are in English, and words like 'ki', 'door', etc. are Hindi words which are transliterated to English. Hindi is the most spoken language in India and fourth most spoken in the world whereas English is the third most spoken language in the world. Thus there are a lot of people who express themselves on social media using English-Hindi code-mixed texts which makes sarcasm detection in such texts much more important.

Several experiments of sarcasm detection have been performed on tweets in English [5],[2],[11] as well as in other languages such as Czech [6], Dutch [7] and Italian [8] but there have been no experiments on English-Hindi code-mixed texts mainly because of the lack of annotated resources.

The main contribution of this paper is to provide a resource of English-Hindi code-mixed tweets which contain both sarcastic and non-sarcastic tweets. We provide tweet level annotation for presence of sarcasm and token level language annotation. This corpus can be used to train, develop and also evaluate the performances of sarcasm detection and language identification techniques on a code-mixed corpus. In addition, we present a baseline supervised classification system for sarcasm detection developed using the same corpus.

Both the dataset and classification system are available online[2].

## 2 Dataset

### 2.1 Data Collection

To collect sarcastic tweets we extract tweets containing hashtags #sarcasm and #irony [9] using the Twitter Scraper API and manually select English-Hindi code-mixed tweets from them. We also use other keywords such as 'bollywood', 'cricket' and 'politics' to collect sarcastic tweets from these domain. Out of these collected tweets, sarcastic and non-sarcastic tweets are further manually separated. To collect more non-sarcastic tweets we extract tweets with keywords such as 'bollywood', 'cricket' and 'politics' which do not contain hashtags #sarcasm and #irony. Further English-Hindi code-mixed tweets are manually selected from them. Having only sarcastic or only non-sarcastic tweets from a particular domain may lead to an unbiased classification system therefore we make sure that there are both sarcastic and non-sarcastic tweets from each domain. The twitter scraper API collects each tweet in json format after which we extract the tweet content and tweet id from it. Figure 1. shows an example of a tweet collected in json format.

[2] https://github.com/sahilswami96/SarcasmDetection_CodeMixed

### 2.2 Data Processing and Annotation

Tweets are annotated by a group of people fluent in both English and Hindi. Each tweet is manually annotated for presence of sarcasm. Tweets are then tokenized and each token is annotated with a language which is manually verified. We used Cohen's Kappa [16] as a measure of inter-annotator agreement and it was calculated to be 0.79.

**Sarcasm Annotation** Each tweet is manually annotated for presence of sarcasm using the tags 'YES' and 'NO'. Tweets with the hashtags #sarcasm and #irony are more likely to contain sarcasm. Tweets which do not contain these hashtags are then manually verified to not contain sarcasm. An example of a tweet (with translation in English) that contains sarcasm and one that does not:

*Tweet: @bonda0123 sir g .. #insomniac likhte ho aur jaldi sone ki baat bhi karte ho !! #irony !!*
*Translation: @bonda0123 sir You write #insomniac and talk about sleeping early !! #irony !!*
Sarcasm: YES

*Tweet: Bhai kuchh bhi karna iss @SimplySajidK ke saath movie mat karna..Bollywood se nafrat ho jaati hai..Itni sadi hui ghatiya filmein banata h ye Translation: Brother do anything but don't do a movie with @SimplySajidk..I start hating Bollywood..They make such bad films*
Sarcasm: NO

Hashtags #sarcasm and #irony are randomly removed from some tweets which contain sarcasm so that the dataset contains both types of sarcastic and ironic tweets, ones with the hashtags #sarcasm and #irony and ones without.

**Tokenization and Language Annotation** There have been several experiments of language identification [10],[13] on various types of texts which motivates the task of token level language annotation in this dataset. Each tweet is tokenized using white spaces as delimiters and taking into account the trends found in the dataset such as use of multiple consecutive punctuations, mentions, etc. Each token is annotated with a language tag. One of the following tags is assigned for language: 'en', 'hi' and 'rest', where 'en' stands for English, 'hi' for Hindi and 'rest' for punctuations, emoticons, named entities, URLs, etc. 'en' is assigned to English words such as 'play', 'warm', etc. and 'hi' is assigned to Hindi words transliterated in English such as 'sahi', 'kya'. Initially each token is assigned language tags using online dictionaries such as Enchant and the 'rest' tags are assigned by identifying hashtags, URLs and mentions. Every language tag and token is manually verified to correct any mistakes in language tags and tokenization. An example of a tweet with language tags:

| Token | Language |
|---|---|
| bhai | hi |
| triple | en |
| talaq | hi |
| se | hi |
| aap | hi |
| kya | hi |
| samjhte | hi |
| hai | hi |
| samjhaye | hi |
| aap | hi |
| zara | hi |
| .. | rest |
| agar | hi |
| triple | en |
| talaq | hi |
| pta | hi |
| hota | hi |
| apko | hi |
| toh | hi |
| aisa | hi |
| nhi | en |
| kehte | hi |
| .. | rest |

**Table 1.** A tweet with token level language annotation

### 2.3 Dataset analysis

The dataset consists of 5250 English-Hindi code-mixed tweets out of which 504 tweets are marked as sarcastic and ironic. The dataset consists of two types of tweets: 1.) Tweets that are marked as sarcastic but do not have hashtags #sarcasm or #irony present in them. 2.) Tweets that contain these hashtags but are not marked as sarcastic. This sparsity in the corpus also helps in developing a better system for sarcasm detection. The average length of a tweet is 22.2 tokens per tweet. The average number of tokens per tweet annotated with 'en', 'hi' and 'rest' tags are 2.1, 16.1 and 4.0 respectively. Table 2. and Table 3. show corpus level and tweet level statistics respectively.

| Category | Number of tweets |
|---|---|
| Total tweets | 5250 |
| Sarcastic tweets | 504 |
| Non-sarcastic tweets | 4746 |

**Table 2.** Corpus level statistics

| Category | Number of tokens |
|---|---|
| Avg. tokens | 22.2 |
| Avg. en tokens | 2.1 |
| Avg. hi tokens | 16.1 |
| Avg. rest tokens | 4.0 |

**Table 3.** Tweet level statistics

### 2.4 Dataset structure

The corpus is structured into three files. The first file contains a tweet id followed by the corresponding tweet text and a blank line and so on. The second file consists of tweet ids followed by language annotated tweets as depicted in Table 1. The third file has the annotation for presence of sarcasm for each tweet. Each tweet id is followed by one of the sarcasm label, a blank line.

## 3 Sarcasm detection system

We present a baseline classification system for sarcasm detection in English-Hindi code-mixed tweets using various word based and character based features. We run and compare various machine learning models which use these features to detect sarcasm.

### 3.1 Preprocessing

It is a common practice on social media to use camel case while writing hashtags. Thus we extract the hashtags from each tweet and extract separate tokens from it by removing the '#' and using a hashtag decomposition approach [16] assuming it is written in camel case. For example we can get 'I', 'Am' and 'Sarcastic' from '#IAmSarcastic'. URLs, mentions, stop words and punctuations are removed from tweets for further processing.

### 3.2 Features

**Word N-Grams** Word n-gram refers to presence or absence of contiguous sequence of n word or tokens in a tweet. Word n-grams have proven to be useful features for sarcasm detection in previous experiments [11],[2],[6]. We consider all n-grams for values of n ranging from 1 to 5. We consider only those n-grams for features which occur at least 10 times in the corpus in order to prune the feature space.

**Character N-Grams** Character n-gram refers to presence or absence of contiguous sequence of n characters in a tweet. It can be observed from previous experiments [2],[6] that character n-grams play an important role in sarcasm detection. We consider all n-grams for values of n ranging from 1 to 3. If we

include all these character n-grams then it will increase the size of the feature vectors enormously thus we consider only those n-grams which occur at least 8 times in the dataset.

**Sarcasm Indicative Tokens** This feature refers to the presence or absence of sarcasm indicative tokens. We use a variation of the approach [14] to find indicative hashtags and extract sarcasm indicative tokens for each language label. We calculate a score for each token where score is defined as:

$$Score(token) = max_{label \in Sarcasm-Set} \frac{freq(token, sarcasm\_label)}{freq(token)}$$

where Sarcasm-Set = {YES, NO}.

We consider only those tokens as features for sarcasm indication which have a score $\geq 0.6$ and occur at least five times in the dataset. We find such tokens for each of the language tags and consider them in the feature vector. The threshold value for scores and number of occurrences has been decided after empirical fine tuning.

**Emoticons** This feature refers to the presence or absence of various emoticons in the tweet. There have been several experiments [5],[6] where emoticons are used as a feature for sarcasm detection. We consider a set of 27 emoticons as features.

### 3.3 Feature Selection

It has been observed in various experiments [12],[15] that feature selection algorithms improve the performance of machine learning models significantly. We use chi square feature selection algorithm which uses chi-squared statistic to evaluate individual feature with respect to each class. This algorithm was used in order to extract the best features and reduce the feature vector size to 500.

### 3.4 Classification Approach

We use three classification techniques: Support Vector Machine with Radial Basis Function kernel, Linear Support Vector Machine, and Random Forest classifier. We use scikit-learn implementation of these methods for sarcasm detection. We also perform 10-fold cross validation on the corpus created to develop the system. 10-fold cross validation is run for each of the individual features separately to observe the effect of each feature on classification.

### 3.5 Results

We use F-score measure to evaluate the performance of our system as the number of sarcastic tweets is less than the number of non-sarcastic tweets and thus using

just accuracy for evaluation of the system may not be a good metric. F-score is defined as the harmonic mean of precision and recall.

$$F - score = 2\frac{precision * recall}{precision + recall}$$

Precision and recall are defined as:

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

where *tp*, *fp* and *fn* are true positives, false positives and false negatives respectively.

Our system achieves a best average F-score of 78.4 after running 10-fold cross validation using the random forest classifier on the dataset. Table 2. shows the F-scores achieved by each of the systems for each feature separately as well as with all the features combined. It can be observed that each feature affects each technique differently. Word n-grams perform best with random forest classifier whereas character n-grams with RBF kernel SVM and sarcasm indicative tokens perform best with linear svm.

| **Features** | **RBF Kernel SVM** | **Random Forest** | **Linear SVM** |
|---|---|---|---|
| Character n-grams | 73.1 | 75.0 | 66.4 |
| Word n-grams | 71.4 | 76.7 | 68.0 |
| Sarcasm indicative tokens | 66.1 | 72.0 | 70.2 |
| Emoticons | 62.8 | 68.5 | 65.7 |
| All features | 76.5 | 78.4 | 71.7 |

**Table 4.** F-scores for all the three classifiers

## 4 Conclusion

With the increase in number of people using social media to express their views, tasks like opinion mining and sentiment analysis have gained a lot of importance. And using sarcasm in these social media texts make these tasks much more challenging.

We presented the first English-Hindi code-mixed dataset for sarcasm detection collected from twitter. We explained the methods used for collecting and annotating these tweets at both tweet level for presence of sarcasm as well as at token level for language. We also presented a baseline supervised classification developed using the same dataset which uses three different machine learning techniques and 10-fold cross validation.

## 5 Future work

This dataset can further be normalized at token level which will thus improve the performance of the classification system. This dataset can also be used to develop systems for automatic language identification in code-mixed texts.

Similar datasets can be created for different language pairs with the presene of other emotions such as humor.

The provided classification system can be improved further by using various other features such as word embeddings, POS tags and other language based features.

## References


1. Aditya Joshi, Pushpak Bhattacharyya, Mark James Carman. 2016. Automatic Sarcasm Detection: A Survey. In *CoRR*(2016).
2. M. Bouazizi, T. Otsuki Ohtsuki. A Pattern-Based Approach for Sarcasm Detection on Twitter. In *IEEE Access, vol. 4, pp. 5477-5488, 2016.*
3. Pooja Deshmukh, Sarika Solanke. 2017. Review Paper: Sarcasm Detection and Observing User Behavioral. In *International Journal of Computer Applications 166(9):39-41, May 2017.*
4. Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, Monojit Choudhury. POS Tagging of English-Hindi Code-Mixed Social Media Content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).*
5. S.K. Bharti, B. Vachha, R.K. Pradhan, K.S. Babu, S.K. Jena. Sarcastic sentiment detection in tweets streamed in real time: a big data approach. In *Digital Communications and Networks, Volume 2, Issue 3, 2016, Pages 108-121*(2016).
6. Tomás Ptácek, Ivan Habernal, Jun Hong, Tomás Hercig. Sarcasm Detection on Czech and English Twitter. In *COLING*(2014).
7. Christine Liebrecht, Florian Kunneman, Antal Van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*(2013).
8. C. Bosco, V. Patti, A. Bolioli. Developing Corpora for Sentiment Analysis: The Case of Irony and Senti-TUT. In *IEEE Intelligent Systems, vol. 28, no. 2, pp. 55-63, March-April 2013.*
9. Erik Forslid, Niklas Wikén. Automatic irony- and sarcasm detection in Social media. 2015.
10. Utsab Barman, Amitava Das, Joachim Wagner, Jennifer Foster. Code Mixing: A Challenge for Language Identification in the Language of Social Media. 2014.
11. David Bamman, Noah Smith. Contextualized Sarcasm Detection on Twitter. In *International AAAI Conference on Web and Social Media*(2015).
12. Sahil Swami, Ankush Khandelwal, Manish Shrivastava, Syed Sarfaraz Akhtar. LTRC IIITH at IBEREVAL 2017: Stance and Gender Detection in Tweets on Catalan Independence. 2017.
13. Amitava Das, Björn Gambäck. Code-Mixing in Social Media Text: The Last Language Identification Frontier?. In *TAL 54: 41-64*(2013).

14. Saif M. Mohammad, Parinaz Sobhani, Svetlana Kiritchenko. Stance and Sentiment in Tweets. In *CoRR*(2016).
15. Can Liu, Wen Li, Bradford Demarest, Yue Chen, Sara Couture, Daniel Dakota, Nikita Haduong, Noah Kaufman, Andrew Lamont, Manan Pancholi, Kenneth Steimel, Sandra Kübler. IUCL at SemEval-2016 Task 6: An Ensemble Model for Stance Detection in Twitter. In *SemEval@NAACL-HLT*(2016).
16. Belainine Billal, Alexsandro Fonseca, Fatiha Sadat. Named Entity Recognition and Hashtag Decomposition to Improve the Classification of Tweets. In *NUT@COLING*(2016).
16. Joseph L. Fleiss, Jacob Cohen. The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability. In *Educational and Psychological Measurement*(1973).