[1. Sarc-M: Sarcasm Detection in Typo-graphic Memes](#_Toc46094472)

[2. Code Mixing: A Challenge for Language Identification in the Language of Social Media](#_Toc46094473)

[3. A Corpus of English-Hindi Code-Mixed Tweets for Sarcasm Detection](#_Toc46094474)

[4. BHAAV- A Text Corpus for Emotion Analysis from Hindi Stories](#_Toc46094475)

[5. Multi-Rule Based Ensemble Feature Selection Model for Sarcasm Type Detection in Twitter](#_Toc46094476)

[6. Natural Language Processing Based Features for Sarcasm Detection: An Investigation Using Bilingual Social Media Texts](#_Toc46094477)

[7. The perfect solution for detecting sarcasm in tweets #not](#_Toc46094478)

# Sarc-M: Sarcasm Detection in Typo-graphic Memes

**WIIFM**: Data acquisition tech. OCR tool, Data cleaning techniques, Tokenization Tech, Feature engineering techniques, models used

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
| 36 | 1 | Sarc-M: Sarcasm Detection in Typo-graphic Memes  By: ***Akshi Kumar, Geetanjali Garg***  Conference Paper  2019 | Typo-graphic images called Memebank scrapped from Instagram  1200 images = 600 typo-graphic images with hashtags #sarcastic, #sarcasm and #irony and negative class data + 600 images with hashtags #motivational and #inspirational. | **Work Summary**   * Data Acquisition from Instagram * Text Extraction using OpenCV * Text Analysis using lexical, pragmatic, semantic features   **Algorithm**   * Modeling using KNN, SVM, RF, DT, MLP   Gap : Only for English Language | **Preprocessing**   * Removal of stop words, placeholders, mentions etc. * Replacement of URLs, special characters such as @, #. * Use of Natural Language tool-Kit (NLTK)\* for tokenization. * Use of Porter's stemmer for stemming to the root word. * Removal of non-ASCII English character. * Part-of- Speech tagging is also done to extract common structural patterns such as verb, adverb, adjective and noun. * Although all the punctuation marks in a tweet are removed as the part of the cleaning process; however, the count of each punctuation mark is kept as we use them as pragmatic features to train the model. In this work, five punctuation-based features that represent figurative text and provide symbolic clues within the tweet are used. These include exclamation marks (!), question marks (?), periods (.), capital letters and use of “or”.   **Feature Eng.**   * Lexical (Top 200 TF-IDF)- Bhatia & Kumar, 2008; Jain et al., 2019] measure is used and the top 200 entries are filtered * Semantic features (sentiment based)   + These lists of words are created using the pos\_tag library under NLTK [Loper & Bird, 2002].   + This lists, for each textual content the number of positive words (pw) and negative words (nw) are counted. emotional positive, negative terms using pos\_tag NLTK   + The adjectives, adverbs and verbs have higher emotional content as compared to nouns, therefore, all the words, either positive or negative that have the associated POS tag, are counted another time to create two additional features that are represent the number of highly emotional positive terms (PW) and highly emotional negative terms(NW) respectively   + ρ(t)=((δ∙PW+pw)−(δ∙NW−nw))⁄((δ∙PW+pw)+(δ∙NW−nw)) * Pragmatic (punctuation based)   + rep: frequency of repetitive alphabets, that is, if alphabet repetition > 2, then set feature to true else false   + excl: frequency of exclamation marks   + ques: frequency of question marks   + dots: frequency of dots   + caps: frequency of capital letters   + quotes: frequency of ‘’ or “” | Acc: 88 using MLP |

# Code Mixing: A Challenge for Language Identification in the Language of Social Media

**WIIFM**: Data acquisition tech. Tagging techniques, Tokenization Tech, Feature engineering techniques, models used, Tools used for Tagging

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
| 40 | 2 | Code Mixing: A Challenge for Language Identification in the Language of Social Media | A Facebook group and 11 Facebook users (20-30 years age group) (known to the authors) were selected to obtain publicly available posts and comments. The Facebook graph API explorer was used for data collection. Since these Facebook users are from West Bengal, the most dominant language is Bengali (Native Language), followed by English and then Hindi (National Language of India). The posts and comments in Bengali and Hindi script were discarded during data collection, resulting in 2335 posts and 9813 comments | **Work Summary**  Automatic language identification for the language of social media. Languages (BN, HI, EN).   * Corpus Acquisition * **Annotation**: Three kinds of annotation: 1- NE, 2- Four tags Sentence/fragment/ Inclusion/WLCM, 3- Six Lang Tag. * **Four basic tags** (viz. sentence, fragment, inclusion and wlcm (word-level code mixing)) to annotate different levels of code mixing. * **Six Lang tag:**  English (en), Bengali (bn), Hindi (hi), Mixed (mixd), Universal (univ) and Undefined (undef). The attribute univ is associated with symbols, numbers, emoticons and universal expressions (e.g. hahaha, lol). The attribute undef is specified for a sentence or a word for which no language tags can be attributed or cannot be categorized as univ. * Named entities annotation. * Handling - Ambiguous Word * Handling - Same word across language * Dictionary Based Detection * Language identification with contextual clue   **Algorithm**   * Word level classification using SVM * Sequence labeling using CRF   Gap: Applicable only for words written in Roman script | * **Tools Used**: Dictionaries: 1- British National Corpus (BNC). 2- SEMEVAL 2013 Twitter Corpus (SemevalTwitter). 3. Lexical Normalization List (LexNorm-List) * **ML Toolkit:** 1- WEKA: We use the Weka toolkit (Hall et al., 2009) for our experiments in decision tree training. 2. MALLET: CRF learning is applied using the MALLET toolkit (McCallum, 2002). 3- Liblinear: We apply Support Vector Machine (SVM) learning with a linear kernel using the Liblinear package (Fan et al., 2008). * **NLP Tools** For data tokenization we used the CMU Tweet-Tokenizer (Owoputi et al., 2013). * **Feature Types:** Word level classification without contextual clue (char-n-gram/G, Presence in dictionary/D, Length of word/L, Capitalization/C | Acc: 95% using SVM |

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

**WIIFM**: Data acquisition tech. Data cleaning techniques, Tokenization Tech, Feature engineering techniques, models used

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
| 38 | 3 | A Corpus of English-Hindi Code-Mixed Tweets for Sarcasm Detection  By: Sahil Swami, Ankush Khandelwal, Vinay Singh, Syed Sarfaraz  Akhtar and Manish Shrivastava  Year: May’2018  **Conference**: Language Technologies Research Centre, International Institute of Information  Technology, Hyderabad | * 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. * The corpus is structured into 3 files.   + tweet id followed by the corresponding tweet text   + tweet ids followed by language annotated tweets   + tweet ids and presence of sarcasm for each tweet. | **Work Summary**   * Data Acquisition:   + Extracted sarcastic tweets containing hashtags #sarcasm and #irony using the Twitter Scraper API. Domain: `bollywood', `cricket' and `politics'   + Extracted non-sarcatic tweets that down cotain #sarcasm, #irony from `bollywood', `cricket' and `politics' hangdles. * Labelling: Sarcasm/Non-sarcasm * Tokenization & Language Annotation (hi, en, rest)   + English words are assigned "en"   + Hindi words typed in Roman assigned "hi"   + url and related assigned "en"   **Algorithms**   * We use three classification techniques: SVM with Radial Basis Function kernel, Linear Support Vector Machine, and Random Forest classifier. * 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.   **Gap:** Only for Roman script. No experiment with Hinglish, the most spoken language of India. No experiment with Transformers. Only 27 emoticons are used. | **Preprocessing:**   * # is removed from hashtag and it is decomposed (assuming camel casing) * URLs, mentions, stop words and punctuations are removed   **Feature Engineering**   * **Word N-Grams**. 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**. 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**. We calculate a score for each token where score is * where Sarcasm-Set = {YES, NO}. We consider only those tokens as features for sarcasm indication which have a score >= 0.6 and occur at least 5 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** : We consider a set of 27 emoticons as features. * **Feature Selection**: 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. | F1: 78% Random Forest |

# BHAAV- A Text Corpus for Emotion Analysis from Hindi Stories

**WIIFM**: Data cleaning techniques, Tokenization Tech, Feature engineering techniques, models used

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  | 4 | BHAAV- A Text Corpus for Emotion Analysis from Hindi Stories  By Yaman Kumar, Debanjan Mahata, Bloomberg LP, Sagar Aggarwal, Anmol Chugh, Rajat Maheshwari, Rajiv Ratn Shah  Corporate Paper (adobe, IIITD, NSIT, USICT, Bloomberg) | The corpus consists of 20,304 sentences collected from 230 different short stories spanning across 18 genres such as Inspirational, Mystery. Each sentence has been annotated into one of the five emotion categories (anger, joy, suspense, sad, and neutral), by three native Hindi speakers with at least ten years of formal education in Hindi. | **Work Summary**   * Data Collection * **Annotation** * Five emotions: *anger*, *joy*, *suspense*, *sad*, and *neutral*. * The extracted text from 230 stories was split into sentences in an automated way * Contained many unnecessary text that were not a part of the story. During the annotation process, the annotators filtered the unwanted text and only annotated the relevant portion. * Whenever the sentences were not correctly split, the annotators also corrected them. * A total of five annotators were used for annotating the entire corpus, such that each sentence gets at-least three annotations. During the annotation process the annotators had access to the actual online story and the list of audio books. Each story was annotated in one sitting. It took **nine** months to finish the process. * एक दिवसीय क्रिकेट मैच में भारत से हार गया पाक. An Indian annotator is often inclined to mark it as *joy* while a Pakistani annotator often marks it as *sad* where as an unbiased reader would read it as having *neutral* emotion. Thus, the annotators were asked to identify only the emotion that an unbiased narrator/reader of that story would like to express while reading it to someone. * **Training**: The BHAAV dataset was randomly shuffled and split into train and test datasets with a ratio of 10:1. But, proportions of labels is same in test and train. 10-fold cross validation was used for the classic techniques. For the deep learning models, random search, was used for selecting the best hyperparameters. The best fitted a fixed randomly selected validation data comprising of 20% of the training data. Only 100 iterations of random search was performed. Once the hyperparameter tuning was done the final model was trained on the entire training data using the selected hyperparameters.   **Preprocessing**   * We tokenize each sentence into words and remove punctuations. * Didn’tt remove the stopwords.   **Algo Used**   * SVM with linear kernel, LogR, RF * Shallow CNN + Birectional LSTM * CLTK for tokenization   **Contribution**   * Annotated Hindi corpus (BHAAV) for sentiment analysis. 20,304 sentences from 230 popular, 18 popular genres. 5 emotion: anger, joy, suspense, sad, and neutral. * A model for Bhaav classification   Gap: It is for pure Hindi language and Devanagari script. Results can be improved with modern transformers. Better features could have been created. | **Feature Engineering**   * Since we deal with Hindi, the standard word tokenizers that are suitable for English language could not be used. Therefore, we used the tokenizer shipped with **Classical Language Toolkit**. * Unigrams, Bigrams and Trigrams were generated as features for each sentence and their TF-IDF (Aizawa, 2003) scores were considered as the feature values. * Since the dataset on which we train our models is relatively small, we use the pretrained word embeddings in order to prevent overfitting. (transfer learning). Fasttext is possibly a better choice than other popular word embedding methods as it is more suitable for representing words belonging to morphologically rich languages like Hindi. * While training the deep learning models, each sentence in the training and test dataset is converted to a fixed size document of 126 words (maximum length of a sentence in the dataset). * Padding is used for sentences of length lesser than 126 words. * Each word is represented as a 300 dimensional (D) vector by the word embedding model. * All the words in the dataset are mapped to their corresponding word embedding vector. Whenever a word is not found in the vocabulary of the word embedding model we assign it a 300-D zero vector. | Accu: 62% with LogR |

# Multi-Rule Based Ensemble Feature Selection Model for Sarcasm Type Detection in Twitter

**WIIFM**: Data extraction techniques, Data cleaning techniques, Feature engineering techniques, Classification models used

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  | 5 | Multi-Rule Based Ensemble Feature Selection Model for Sarcasm Type Detection in Twitter  By Karthik Sundararajan and Anandhakumar Palanisamy  Computational Intelligence and Neuroscience  Year Jan’2020 | 76799 English tweet kept and remaining were removed. | **Work Summary**   * Data Acquisition using Twitter API (tweepy and twython) based on hashtag * Data Cleaning * Feature Extration * Classification into sarcasm, non-sarcasm * Feature ensembling has been proposed to identify the optimal set of features for classifying sarcasm into various types, the best set of features that provide a better accuracy for the aforementioned classifiers is selected for determination of type of sarcasm. Features are then grouped into various categories such as linguistic features, contradictory features, and sentiment-based features.   **Contribution**   * Classify sarcasm into various types based on the emotional aspects * Determine optimum set of feature for classification * Propose a multi-rule based approach for classification to handle vagueness, uncertainty.   **Preprocessing**   * Hashtags is tokenized * URLs, @, and links are converted into meaning text * RT, space, punctuation removed. * POS tagging, stemming, and lemmatization are performed to obtain understandable data.   **Models**  Random Forest, Naive Bayes, Support Vector Machine, K-Nearest Neighbor, Gradient Boosting, AdaBoost, Logistic Regression, and Decision Tree. | **Feature Engineering**   * Lexical features include *n*-gram, bigram, and unigram which are combination of words that are extracted from the tweets to aid in tokenization. * Intensifiers are also identified as they might help in the sarcasm detection process. * Pragmatic features like emoticon and smileys are extracted. * The proposed system extracts a total of 20 features: noun and verb count positive intensifier, negative intensifier, bigram, trigram, skip gram, unigram, emoji sentiment, sentiment score, interjections, punctuators, exclamations, question mark, uppercase, repeat words count, positive word frequency, negative word frequency, polarity flip, and parts of speech tagging * Uppercase words are extracted as features because sometimes people use capital lettered words to stress on the things that they want to convey strongly | F1 87.4 ensamble |

# Natural Language Processing Based Features for Sarcasm Detection: An Investigation Using Bilingual Social Media Texts

**WIIFM**: Data extraction (from FB) techniques, Feature engineering techniques, evaluating models with different number of features, use of idioms/figure of speech

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  | 6 | Natural Language Processing Based Features for Sarcasm Detection: An Investigation Using Bilingual Social Media Texts  By: Mohd Suhairi, Md Suhaimin, Mohd Hanafi Ahmad Hijazi, Rayner Alfred and Frans Coenen  8th International Conference on Information Technology (ICIT) | * The bilingual corpus used for evaluation purposes was acquired from comments related to economic news from Facebook public pages. * The comments were acquired using Graph API Explorer12 and the Facebook Query Language (FQL). All links, pictures and video were filtered from the data. Thus only text based comments were considered. A total of 3000 comments were acquired and annotated manually, according to whether they featured sarcasm or not, by three annotators. * The annotations were considered to be “valid” only if all three annotators agreed. In this manner a subset of 1970 comments was derived from the original set of 3000 comments, 969 sarcastic comments and 1001 non-sarcastic comments. | Contribution:   * process for extracting features indicative of the presence of sarcasm in bilingual social media texts and * set of NLP based feature categories appropriate for sarcasm detection in the context of Malay social media data   Work Summary:   * *Data Collection from FB* * *Feature Extraction* * *Extraction from Bilingual Corpus*   + *Preprocessing of the Corpus and Lexical Features Extraction*   + *Pragmatic Features Extraction*   + *Prosodic (Malay) Features Extraction* * *Extraction of Features from English Translated Corpus* * *Corpus Translation to English using google* * *Prosodic (English) Features Extraction and Combination* * *Syntactic Features Extraction* * *Idiosyncratic Features Extraction* * *SVM Linear Classification*   Gap: Used only for Malay. Used only Roman script. Used translation from Malay word Roman script to English translation. It will cause of lots of mistake translation due to 2 reason a- pronunciation of actual word and roman letter used will be different. b- loss of meaning. | * *Extraction from Bilingual Corpus*   + *Preprocessing of the Corpus and Lexical Features Extraction:*     - tokenization, spellchecking, remove duplicate.     - used Malay and English dictionaries to correct misspelled words.     - used Malay and English stopword lists for our bilingual data.     - Lexical feature were then extracted from the corpus in the form of n-grams.  Single character such as ‘n’, ‘t’, and ‘b’ were omitted.     - lowercased all the tokens.   + *Pragmatic Features Extraction:*     - Punctuation marks were considered to be pragmatic features, instead of sentence segmentators, because of their potential to indicate sarcasm.     - question marks (?), exclamation marks (!) and quotation marks (“” and ‘’). In addition hashtags (#) also considered.     - The length of sequences of punctuation marks was reduced to a maximum of three characters to avoid dispersion.   + *Prosodic (Malay) Features Extraction:*     - In the third step a Malay list of interjections7 was employed. It should be noted that interjections differs according to language, for example “*ooi*”, “*puii*” and “*weii*” are only found in Malay. A total of 43 Malay interjections were identified and used.     - Malay prosodic features were thus extracted from the preprocessed corpus using this Malay interjection list. * *Extraction of Features from English Translated Corpus*   + *Corpus Translation to English using google translate.*The translation preserved some Malay words such as names, locations and abbreviations   + *Prosodic (English) Features Extraction and Combination.*English prosodic features were then extracted from the translated corpus using the English interjection’s list   + *Syntactic Features Extraction.*We choose four groups of POS: NOUN, VERB, ADJECTIVE and ADVERB. From the translated corpus, we tagged each token using the Penn Treebank POS tagset built up of 36 different tags.   + *Idiosyncratic Features Extraction:*Motivated by study in the field of linguistics, we created a syntax rule in the form of NOUN-ADPOSITION-NOUN to identify idiosyncratic phraseology from the corpus. For example, the phrase ‘head of cabbage’ was identified as an idiosyncratic phrase as the ‘head’, ‘of’ and ‘cabbage’ will be tagged as noun, adposition and noun by the POS tagger. | F1=85 if Syntactic + Pragmatic + Prosodic + Idiosyncratic features used. |

# The perfect solution for detecting sarcasm in tweets #not

**WIIFM**: Nothing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  | 7 | The perfect solution for detecting sarcasm in tweets #not | * 8K Dutch tweets with this hashtag. Assuming that the human labeling is correct (annotation of a sample indicates that about 85% of these tweets are indeed sarcastic), * we train a machine learning classifier on the harvested examples, and apply it to a test set of a day’s stream of 3.3 million Dutch tweets. | Nothing Speicific | Nothing Special | AUC = 75% |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  |  | Harnessing Online News for Sarcasm Detection  in Hindi Tweets |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  |  | Hindi Part of Speech(POS) Tagging |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  |  | Sentiment Analysis on Swachh Bharat using Twitter |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  |  | Sentiment Analysis of Hindi Reviews based on Negation and Discourse Relation |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ref** | **Sno** | **Title / Author/ Year/ Article Type** | **Dataset (Not available in public domain)** | **Work Summary (WS) /Problem Statement (PS)/ Future Scope (FS)/ Algo** | **Preprocessing / Feature Engineering** | **Evaluation** |
|  |  | Sentiment Analysis in a Resource Scarce Language:Hindi |  |  |  |  |

**Papers to read for Sarcasm Detection in Hinglish Language (SDHL) Project**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sno. | Filename | Type | Sarcasm (Yes/No) | To Read (Yes/No) | Read(Yes/No) | Summarised (Yes/No) |
| 1 | Semi-Supervised Mix-Hindi Sentiment Analysis using Neural Network |  |  | Y |  |  |
| 2 | Medium: NLP (Sentiment Analysis) — Hindi!!! |  |  | Y |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 | Context-based Sarcasm Detection in Hindi Tweets |  |  | Y |  |  |
| 7 | Sentiment Analysis For Hindi Language |  |  | Y |  |  |
| 8 | Lexicon-Based Sentiment Analysis in the Social Web |  |  | Y |  |  |
| 9 | Sentiment Analysis For Hindi Language |  |  | Y |  |  |
| 10 | Semi-Supervised Mix-Hindi Sentiment Analysis using Neural Network |  |  | Y |  |  |
| 11 | BHAAV- A Text Corpus for Emotion Analysis from Hindi Stories |  | Y | Y | Y | Y |
| 12 | Aspect based Sentiment Analysis in Hindi: Resource Creation and Evaluation |  |  | Y |  |  |
| 13 | CARER: Contextualized Affect Representations for Emotion Recognition |  |  | Y |  |  |
| 14 | Real-time Sentiment Analysis of Hindi Tweets |  |  | Y |  |  |
| 15 | Fake News Detection on Social Media: A Data Mining Perspective |  |  | Y |  |  |
| 16 | WiLI-2018 - Wikipedia Language Identification database |  |  | Y |  |  |
| 17 | Twitter Sentiment Analysis Training Corpus (Dataset) |  |  | Y |  |  |
| 18 | NLP (Sentiment Analysis) — Hindi!!! |  |  | Y |  |  |
| 19 | Recent Trends in Deep Learning Based Natural Language Processing (Jul2018) - Review Article |  |  | Y |  |  |
| 20 | Harnessing Online News for Sarcasm Detection in Hindi Tweets |  |  | Y |  |  |
| 21 | Semantics-aware BERT for Language Understanding |  |  | Y |  |  |
| 22 | Practical guide to Attention mechanism for NLU tasks |  |  | Y |  |  |
| 23 |  |  |  |  |  |  |
| 24 | A Corpus of English-Hindi Code-Mixed Tweets for Sarcasm Detection.pdf |  | Y | Y | Y | Y |
| 25 | A Pragmatic Analysis of Humor in Modern Family.pdf |  |  | Y |  |  |
| 26 | A TENGRAM method based part-of-speech tagging of multi-category words.pdf |  |  | Y |  |  |
| 27 | Allen - 2017 - Maximum Likelihood Estimation.pdf |  |  | Y |  |  |
| 28 | Anant Kumar, Satwinder Singh - 2019 - Fake News Detection of Indian and United States Election Data using Machine Learning Algorithm.pdf |  |  | Y |  |  |
| 29 | Automatic Sarcasm Detection- A Survey.pdf |  |  | Y |  |  |
| 30 | Bakshi et al. - 2016 - Opinion mining and sentiment analysis.pdf |  |  | Y |  |  |
| 31 | Bansal, Advisor, Mukherjee - Unknown - Sentiment Analysis In Hindi.pdf |  |  | Y |  |  |
| 32 | Bhattacharya, Goyal, Sarkar - 2016 - Using Word Embeddings for Query Translation for Hindi to English Cross Language Information Retriev.pdf |  |  | Y |  |  |
| 33 | Chavan - 2018 - Sentiment Classification of News Headlines on India in the US Newspaper Semantic Orientation Approach vs Machine Learnin.pdf |  |  | Y |  |  |
| 34 | CodeMixing-A Challenge for Language Identification in the Language of Social Media.pdf |  | Y | Y | Y | Y |
| 35 | Context-based Sarcasm Detection in Hindi Tweets.pdf |  |  | Y |  |  |
| 36 | Dennis Gitari et al. - 2015 - A Lexicon-based Approach for Hate Speech Detection.pdf |  |  | Y |  |  |
| 37 | Devlin et al. - Unknown - BERT Pre-training of Deep Bidirectional Transformers for Language Understanding.pdf |  |  | Y |  |  |
| 38 | EntityRank-Kavita-Ganesan.pdf |  |  | Y |  |  |
| 39 | Extensions to HMM-based StatisticalWord Alignment Models.pdf |  |  | Y |  |  |
| 40 | FakeNews-Detection-SocialMedia.pdf |  |  | Y |  |  |
| 41 | FakeNewsNet\_A\_Data\_Repository\_with\_News\_Content\_So.pdf |  |  | Y |  |  |
| 42 | Gaikwad, Haribhakta - Unknown - Adaptive GloVe and FastText Model for Hindi Word Embeddings.pdf |  |  | Y |  |  |
| 43 | Harnessing Online News for Sarcasm Detection in Hindi.pdf |  |  | Y |  |  |
| 44 | Hunt Allcott Matthew Gentzkow - 2016 - Social Media and Fake News in the 2016 Election.pdf |  |  | Y |  |  |
| 45 | Indic SentiReview- Natural Language Processing based Sentiment Analysis on major Indian Languages.pdf |  |  | Y |  |  |
| 46 | Investigations in Computational Sarcasm.pdf |  |  | Y |  |  |
| 47 | Irony, Sarcasm and Parody in the American Sitcom.pdf |  |  | Y |  |  |
| 48 | Joulin et al. - 2016 - FastText.zip Compressing text classification models.pdf |  |  | Y |  |  |
| 49 | Kak - 1987 - The Paninian approach to natural language processing.pdf |  |  | Y |  |  |
| 50 | Khan, Anwar, Durrani - 2017 - Machine Translation Approaches and Survey for Indian Languages.pdf |  |  | Y |  |  |
| 51 | Knowledge Representation in Sanskrit-Rick\_Bricks.pdf |  |  | Y |  |  |
| 52 |  |  |  |  |  |  |
| 53 | Lexicon-Based Methods for Sentiment Analysis.pdf |  |  | Y |  |  |
| 54 | Lexicon-BasedSentimentAnalysisintheSocialWeb-JBasicApplSciRes46238-2482014ISSN2090-4304.pdf |  |  | Y |  |  |
| 55 | Liu, Bi, Fan - 2017 - A method for multi-class sentiment classification based on an improved one-vs-one (OVO) strategy and the support v.pdf |  |  | Y |  |  |
| 56 | Machine Learning in Automated Text Categorization.pdf |  |  | Y |  |  |
| 57 | Machine Translation of Bi-lingual Hindi-English (Hinglish) Text .pdf |  |  | Y |  |  |
| 58 | method for multi-class sentiment classification based on an improved-OVO.pdf |  |  | Y |  |  |
| 59 | Mikolov et al. - Unknown - Efficient Estimation of Word Representations in Vector Space.pdf |  |  | Y |  |  |
| 60 | Mikolov, Le - Unknown - Exploiting Similarities among Languages for Machine Translation.pdf |  |  | Y |  |  |
| 61 | Multimodal Markers of Irony and Sarcasm.pdf |  |  | Y |  |  |
| 62 | Natural language processing based features for sarcasm detection- An investigation using bilingual social media texts.pdf |  |  | Y |  |  |
| 63 | Natural\_Language\_Processing\_with\_Python.pdf |  |  | Y |  |  |
| 64 | NLP-Panini.pdf |  |  | Y |  |  |
| 65 | Real-time Sentiment Analysis of Hindi Tweets\_Ghosh\_Dutta.pdf |  |  | Y |  |  |
| 66 | Recent Trends in Deep Learning Based Natural Language Processing.pdf |  |  | Y |  |  |
| 67 | Recognition of consonant-vowel (CV) units under background.pdf |  |  | Y |  |  |
| 68 | ROC Graphs- Notes and Practical Considerations for Researchers.pdf |  |  | Y |  |  |
| 69 | Sanskrit\_Word\_Segmentation\_Using\_Charact.pdf |  |  | Y |  |  |
| 70 | Saravia et al. - 2018 - CARER Contextualized Affect Representations for Emotion Recognition.pdf |  |  | Y |  |  |
| 71 | Sarcasm Detection in Hindi sentences using Support Vector.pdf |  |  | Y |  |  |
| 72 | Sarcasm Detection on Twitter.pdf |  |  | Y |  |  |
| 73 | Sarc-M- Sarcasm Detection in Typo-graphic Memes.pdf |  | Y | Y | Y | Y |
| 74 | Sebastiani - Unknown - Machine Learning in Automated Text Categorization.pdf |  |  | Y |  |  |
| 75 | Semantics-aware BERT for Language Understanding.pdf |  |  | Y |  |  |
| 76 | SentencePiece- A simple and language independent subword tokenizer.pdf |  |  | Y |  |  |
| 77 | Sentiment Analysis of Hindi Review based on Negation and Discourse Relation.pdf |  |  | Y |  |  |
| 78 | Sentiment\_Analysis\_For\_Hindi\_Language\_Final\_Thesis.pdf |  |  | Y |  |  |
| 79 | SentimentAnalysis-Hindi.pdf |  |  | Y |  |  |
| 80 | Sentiment-Analysis-in-a-Resource-Scarce-Language-Hindi.pdf |  |  | Y |  |  |
| 81 | Sharma et al. - 2014 - A Sentiment Analyzer for Hindi Using Hindi Senti Lexicon.pdf |  |  | Y |  |  |
| 82 | Sundararajan, Palanisamy - 2020 - Multi-Rule Based Ensemble Feature Selection Model for Sarcasm Type Detection in Twitter.pdf | Conf | Y | Y | Y | Y |
| 83 | TEXT NORMALIZATION of CODE MIX and SENTIMENT ANALYSIS.pdf |  |  | Y |  |  |
| 84 | The differential role of ridicule in sarcasm and irony, Metaphor and Symbol.pdf |  |  | Y |  |  |
| 85 | The highest form of intelligence= Sarcasm increases creativity for both expressers and recipient.pdf |  |  | Y |  |  |
| 86 | The perfect solution for detecting sarcasm in tweets #not.pdf |  | Y | Y | Y | Y |
| 87 | Thumbs Up or Thumbs Down- Semantic Orientation Applied to Unsupervised Classification of Reviews.pdf |  |  | Y |  |  |
| 88 | Toutanova, Ilhan, Manning - 2002 - Extensions to HMM-based statistical word alignment models.pdf |  |  | Y |  |  |
| 89 | Tutorial-SA-Hindi-GIAN.pdf | PPT |  | Y |  |  |
| 90 | Use of Sanskrit for natural language processing.pdf |  |  | Y |  |  |
| 91 | Vijay et al. - Unknown - A Dataset for Detecting Irony in Hindi-English Code-Mixed Social Media Text.pdf |  |  | Y |  |  |
| 92 | Wawre, Deshmukh - 2013 - Volume 5 Issue 4, April 2016 www.ijsr.net Licensed Under Creative Commons Attribution CC BY Sentiment Classific.pdf |  |  | Y |  |  |
| 93 | SQuAD: 100,000+ Questions for Machine Comprehension of Text |  |  | N |  |  |
| 94 | SG-Net: Syntax-Guided Machine Reading Comprehension |  |  | N |  |  |
| 95 | Retrospective Reader for Machine Reading Comprehension |  |  | N |  |  |
| 96 | [Question Answering on SQuAD2.0 : https://paperswithcode.com/sota/question-answering-on-squad20?p=read-verify-machine-reading-comprehension  https://rajpurkar.github.io/SQuAD-explorer/](https://paperswithcode.com/sota/question-answering-on-squad20?p=read-verify-machine-reading-comprehension) |  |  | N |  |  |
| 97 | Evaluation Metrics for Unsupervised Learning Algorithms |  |  | N |  |  |
| 98 | Unsupervised Learning and Data Clustering |  |  | N |  |  |
| 99 | The 5 Clustering Algorithms Data Scientists Need to Know |  |  | N |  |  |
| 100 | Dr. Amba (Complete Work)- Hyderabad Univ (Study her work) |  |  | N |  |  |
| 101 | Quran Datasets in Different Languages (Hindi) |  |  | N |  |  |
| 102 | Sanskrit as a Programming Language and Natural Language Processing |  |  | N |  |  |
| 103 | Bias correction of numerical prediction model temperature forecast Data Set |  |  | N |  |  |
| 104 | All\_OfStatistics\_Statistical\_Inference\_2004.pdf |  |  | N |  |  |
| 105 | AN ANALYSIS OF CURRENT TRENDS FOR SANSKRIT AS A COMPUTER PROGRAMMING LANGUAGE.pdf |  | N | N |  |  |
| 106 | An Introduction to Statistical Learning\_7th.pdf |  |  | N |  |  |
| 107 | Appen-How-to-Create-an-AI-Center-of-Excellence-for-Enterprise-eBook.pdf |  |  | N |  |  |
| 108 | Approaches to Cross-Domain Sentiment Analysis- A Systematic Literature Review.pdf |  |  | N |  |  |
| 109 | Aslyan - 2011 - Syllable Based Speech Recognition.pdf |  |  | N |  |  |
| 110 | Bathulapalli, Desai, Kanhere - 2016 - Use of Sanskrit for natural language processing.pdf |  |  | N |  |  |
| 111 | BaysianReasoning&MachineLearning.pdf |  |  | N |  |  |
| 112 | BigBook-Experimentation-Main-ebook.pdf |  |  | N |  |  |
| 113 | Breiman - 2001 - ST4\_Method\_Random\_Forest.pdf |  |  | N |  |  |
| 114 | Chen, Guestrin - 2016 - XGBoost A scalable tree boosting system.pdf |  |  | N |  |  |
| 115 | Chen, Guestrin - Unknown - LearningSys\_2015\_paper\_32.pdf |  |  | N |  |  |
| 116 | Daniel, Martin - Unknown - Speech and Language Processing.pdf |  |  | N |  |  |
| 117 | Data\_Science\_from\_Scratch\_Joel\_Grus.pdf |  |  | N |  |  |
| 118 | DataStructuresAndAlgorithmsWit\_2015.pdf |  |  | N |  |  |
| 119 | Deep Learning Based Unsupervised POS Tagging for Sanskrit.pdf |  |  | N |  |  |
| 120 | deep\_learning\_with\_python\_mini\_course.pdf |  |  | N |  |  |
| 121 | DiscreteMathematics\_2003.pdf |  |  | N |  |  |
| 122 | Downey - 2008 - Think Python How to Think Like a Computer Scientist V3.2.pdf |  |  | N |  |  |
| 123 | Downey - 2012 - Think Complexity.pdf |  |  | N |  |  |
| 124 | Dua et al. - Unknown - DROP A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs.pdf |  |  | N |  |  |
| 125 | Dynamic\_Routing\_Between\_Capsules.pdf |  |  | N |  |  |
| 126 | Effect of speech coding on speaker identification.pdf |  |  | N |  |  |
| 127 | Erickson - 1867 - Standard Normal Probabilities.pdf |  |  | N |  |  |
| 128 | example-CLUE Based Load Balancing in Replicated Web.pdf |  |  | N |  |  |
| 129 | Experience Grounds Language.pdf |  |  | N |  |  |
| 130 | Faster R-CNN Towards Real-Time Object.pdf |  |  | N |  |  |
| 131 | FundamentalsOfPowerElectronics\_2001.pdf |  |  | N |  |  |
| 132 | Ganesan, Zhai - Unknown - Opinion-Based Entity Ranking (Author's Draft).pdf |  |  | N |  |  |
| 133 | Going deeper with convolutions.pdf |  |  | N |  |  |
| 134 | Grokking Algorithms.pdf |  |  | N |  |  |
| 135 | Guide2DataMining.pdf |  |  | N |  |  |
| 136 | HandbookOfTheLifeCourse\_2003.pdf |  |  | N |  |  |
| 137 | Hands On Machine Learning with Scikit Learn and TensorFlow.pdf |  |  | N |  |  |
| 138 | He - Unknown - XGBoost.pdf |  |  | N |  |  |
| 139 | HotPotQA.pdf |  |  | N |  |  |
| 140 | Imagenet-classification-with-deep-convolutional-neural-networks.pdf |  |  | N |  |  |
| 141 | introduction to algorithm 2rd edition (MIT Press).pdf |  |  | N |  |  |
| 142 | jurafsky+martin.slp3draft.ch9.pdf |  |  | N |  |  |
| 143 | JustinDyerThesis.pdf |  |  | N |  |  |
| 144 | Ke et al. - 2017 - LightGBM A highly efficient gradient boosting decision tree.pdf |  |  | N |  |  |
| 145 | K-means with Three different Distance Metrics.pdf |  |  | N |  |  |
| 146 | Lecture+Notes+-+Recurrent+Neural+Networks.pdf |  |  | N |  |  |
| 147 | LightGBM A Highly Efficient Gradient Boosting.pdf |  |  | N |  |  |
| 148 | Machine-Learning-Yearning.pdf |  |  | N |  |  |
| 149 | Marina Santini - 2016 - Advantages & Disadvantages of k-Means and Hierarchical clustering (Unsupervised Learning).pdf |  |  | N |  |  |
| 150 | Messaging - Unknown - STREAMING WITH Publish Subscribe Messaging with Kafka.pdf |  |  | N |  |  |
| 151 | Mining\_Of\_Massive\_Datasets.pdf |  |  | N |  |  |
| 152 | NeuralNetwork-neuron-alenetze-en-zeta2-2col-dkrieselcom.pdf |  |  | N |  |  |
| 153 | NeuralNetworksAndDeepLearning\_2018.pdf |  |  | N |  |  |
| 154 | Number Cruncher Statistical Systems (NCSS), Reserved - 2019 - Hierarchical ClusteringDendrograms.pdf |  |  | N |  |  |
| 155 | Openai et al. - Unknown - Improving Language Understanding by Generative Pre-Training.pdf |  |  | N |  |  |
| 156 | Presenting\_Data\_People\_Wont\_Ignore\_Ebook.pdf |  |  | N |  |  |
| 157 | Rajpurkar et al. - 2016 - SQuad 100,000 questions for machine comprehension of text.pdf |  |  | N |  |  |
| 158 | Randomforest-2001.pdf |  |  | N |  |  |
| 159 | Reading+the+abstract.pdf |  |  | N |  |  |
| 160 | Representing Social Media Users for Sarcasm Detection.pdf |  |  | N |  |  |
| 161 | Robinson - 2009 - Time Series Analysis.pdf |  |  | N |  |  |
| 162 | Sabour, Frosst, Hinton - 2017 - Dynamic Routing Between Capsules.pdf |  |  | N |  |  |
| 163 | Sanskrit Sandhi Splitting using seq2(seq).pdf |  |  | N |  |  |
| 164 | Schapire - 1999 - A brief introduction to boosting.pdf |  |  | N |  |  |
| 165 | Simonyan, Zisserman - 2015 - Very deep convolutional networks for large-scale image recognition.pdf |  |  | N |  |  |
| 166 | Sledge - 2017 - Superintelligence.pdf |  |  | N |  |  |
| 167 | SocialAnxietyAndSocialPhobiaIn\_2005.pdf |  |  | N |  |  |
| 168 | Sound Classification Using Convolutional Neural.pdf |  |  | N |  |  |
| 169 | Springer Ebooks.pdf |  |  | N |  |  |
| 170 | Springer-from-DrManoj.pdf |  |  | N |  |  |
| 171 | superintelligence.pdf (Reinforcement Learning: An Introduction) |  |  | N |  |  |
| 172 | SuttonBartoIPRLBook2ndEd.pdf |  |  | N |  |  |
| 173 | The Elements of Statistical Learning.pdf |  |  | N |  |  |
| 174 | Think\_Stats\_Probality\_and\_Stats\_for\_Programmers.pdf |  |  | N |  |  |
| 175 | Understanding-machine-learning-theory-algorithms.pdf |  |  | N |  |  |
| 176 | Unknown - 2011 - Skillful writing of an awful research paper.pdf |  |  | N |  |  |
| 177 | UpGrad - 2018 - Recurrent Neural Network - Upgrad Notes.pdf |  |  | N |  |  |
| 178 | VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION.pdf |  |  | N |  |  |
| 179 | When BERT Plays the Lottery, All Tickets Are Winning.pdf |  |  | N |  |  |
| 180 | XGBoost A Scalable Tree Boosting System.pdf |  |  | N |  |  |
| 181 | XGBoost Reliable Large-scale Tree Boosting System.pdf |  |  | N |  |  |
| 182 | xgboost.pdf |  |  | N |  |  |
| 183 | Zhang, Yang, Zhao - 2020 - Retrospective Reader for Machine Reading Comprehension.pdf |  |  | N |  |  |
| 184 | Zobel - Writing for computer science 3rd edition.pdf |  |  | N |  |  |