Automatic Sarcasm Detection: Reddit Corpus

Name:	**Vatsala Nema**
Registration No./Roll No.:	19332
Institute/University Name:	IISER Bhopal
Program/Stream:	Electrical Engineering and Computer Sciences
Problem Release date:	August 08, 2022
Date of Submission:	September 27 2022

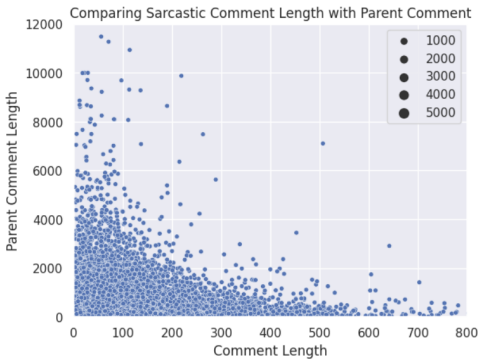
1 Introduction

Through this project we have tried to detect Sarcasm over reddit corpus using Natural Language Processing and various Machine learning based classification techniques. Sarcasm detection becomes ambiguous as individuals state the opposite of what is implied, hence it becomes a very challenging task. Here, we have classified posts on Reddit as being 'Sarcastic' or 'Non-Sarcastic' based on the given labelled training dataset. The training corpus has 8,08,090 Reddit posts which are labelled. Of these 8,08,090 labelled comments , almost half of the comments i.e. '4,04,020' are labelled non-sarcastic and '4,04,011' are sarcastic comments. Thus the dataset is balanced. The corpus was released as part of this publication: A Large Self-Annotated Corpus for Sarcasm.
The task was to develop an NLP method that could identify the sarcastic comments perfectly based on the learnings from the labelled dataset. We used bag of words model and TF-idf vectorizers and applied 3 classifiers namely, Multinomial Naive Bayes, Logistic regression and Support vector machine to train the machine to learn labelling. We applied different NLP techniques like stop-word removal, lemmatization and stemming on the dataset to test for the accuracy of prediction. Later, we used the best trained model to predict the class 'sarcastic' or 'non-sarcastic' on the given test dataset.

2 Methods

2.1 Data Source

Self-Annotated Reddit Corpus (SARC): SARC cite is a collection of labelled sarcastic and non-sarcastic Reddit posts and comments released by Khodak, Saunshi, and Vodrahalli. Sarcasm labelling is performed by adding a trailing "/s" after the post's text on reddit. While sarcasm self-labelling is a common posting convention, a noise component is involved. Not all users abide by this convention, which leads to non-sarcastic labelled sarcastic posts. Moreover, since Sarcasm is ambiguous to detect manual labelling is also not foolproof. When checked against human-annotated labelling, a false negative rate of 2 % and a false positive rate of 1% were determined to exist.



2.2 Exploratory Data Analysis

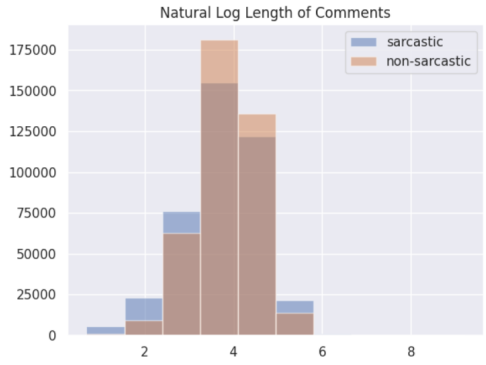
On exploring the data, we found that the number of sarcastic and non-sarcastic comments are equal to each other. To further find meaning in the dataframe, we plot and try to find a relation between the natural log lengths of the sarcastic comments vis-a-vis non-sarcastic comments. As

shown in Figure 2, we can see that the sarcastic comment length is normally distributed but the non-sarcastic comment length is a little right-skewed. Next, we tried to find the relation between comments and parent comments, for that we plotted sarcastic comments and their parent comments and found out that the parent comment length is very large (12000 words) as compared to comment length (800 words). The graph here shows that the Length of parent comments and the length of the comment are having a hyperbolic relation.

2.3 Pre-processing and Feature Engineering

2.3.1 Lemmatization

Lemmatization is the grouping together of different morphological forms(called lemmas) of the same word. it allows end users to query any version of a base word and get relevant results.



2.3.2 Stemming

Stemming is referred to as the practice of stripping off prefixes and suffixes that have been added to the word's base form. This can become quite complicated in languages other than English, whose only inflected forms are singular/plural, verb tense and comparative/superlative forms of adverbs and adjectives.

2.3.3 Stop word removal

Stop word removal is the process of removing commonly occurring words like the, is, and, that are not relevant to our analysis or problem statement and only increase the dimensions of the dataset. Removing stop words enables us to reduce our input data's dimensions and thus helps reduce processing time.

2.3.4 Bag of Words model

Bag of words is a representation of text that describes the occurrence of words within a document. It considers word count as a feature and computes it using a histogram. The phenomenon behind it is that similar documents would have Any information about the order of the words is deleted and it is only concerned with whether known words occur in the document, instead of where they occur. It is highly interpretable and highly flexible in terms of customizing our specific text data.

2.3.5 TF-IDF

Term Frequency-Inverse Document Frequency of Records is a technique to calculate the relevance of the words in terms of how frequent they are in the given text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus. Term-frequency: Defined as the number of times a given word occurs in a document. Document Frequency: Defined as the number of documents the word occurs in. Inverse Document frequency: Ratio of number of documents containing the particular word with the Document Frequency. This term helps normalize the frequency over the entire corpus.

2.4 Classification Techniques

2.4.1 Logistic Regression

Logistic regression model is a regression technique in which the dependent variable i.e., the target is categorical. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes.LR is a transformation of a linear regression using the sigmoid function. The vertical axis stands for the probability for a given classification and the horizontal axis is the value of x. It assumes that the distribution of $y|x$ is Bernoulli distribution. The formula for LR is $F = 1/(1 + e^{-(\beta_0+\beta_1 x)})$ Here $\beta_0 + \beta_1 x$ is similar to the linear model y = ax + b. The logistic function applies a sigmoid function to restrict the y value from a large scale to within the range 0–1.

2.4.2 Multinomial Naive Bayes

Naive Bayes classifier is based on Bayes theorem. They are from the background of probabilistic classifiers with "naive" assumption of conditional independence between every pair of a feature. Multinomial Naive Bayes theorem is widely used in NLP text classification tasks. MNB predicts the tag of a text. They calculate the probability of each tag for a given text and then output the tag with the highest one.

2.4.3 Decision Tree Classifier

It is a supervised learning algorithm. In this algorithm, data are continuously split into smaller parts until it reaches its class. It uses the terminologies like nodes, edges, and leaf nodes. Decision trees are presented similar to a flow chart, with a tree structure wherein instances are classified according to their feature values. A node in a decision tree represents an instance, outcomes of the test represented by branch, and the leaf node epitomized the class label.

3 Experimental Analysis

The classification was performed in four stages using four classifiers using a pipeline to determine the best hyper-parameters: Logistic regression, Multinomial Naive Bayes, decision tree, and random forest. We used the given classifiers because they are commonplace in NLP tasks. All of them are easy to interpret and flexible to change making them very viable to use for our case.

The first stage of pre-processing made use of lexical features processed with Bag-of-Words (BoW) and term frequency inverse document frequency (TF-IDF) to form a dictionary. The second stage involved Stopword Removal. And the third and fourth stages comprised of Stemming and Lemmatization respectively. This dictionary is then run through the three classifiers to determine a prediction label based solely on the vectorized features.We planned on using different combination of the techniques to enhance the final filtered dataset, but could not do so at this stage. We plan on implementing it in the phase-2. The

We can see from the table below that the preprocessing techniques made quite a difference in accuracy which is how we expected it to be since lemmatization, stemming and stop word removal essentially reduce dimensionality and make the dataset more easy to analyze. The evaluation of the classifiers in this project have been done using three criteria, one is micro-averaged precision, recall and f-measure. Micro-averaged Precision is used as an evaluation measure because it treats the entire set of data as an aggregate result, and calculates 1 metric rather than k metrics that get averaged together. Recall is calculated as the number of true positives divided by the total number of true positives and false negatives.

$$Recall = TruePositives/(TruePositives + FalseNegatives)$$

The result is a value between 0.0 for no recall and 1.0 for full or perfect recall. Recall is used as the focus is on minimizing the false negatives. F1 score is used as it summarizes both precision and recall succinctly.

Classifier	Preprocessing Technique	Accuracy	Micro-averaged Precision	Recall	F measure
Logistic Regression	Without Stop word removal	0.595	0.595	0.597	0.588
	With stop word removal	0.594	0.594	0.600	0.578
	Lemmatization	0.6	0.6	0.593	0.612
	Stemming	0.6	0.6	0.5932	0.611
Decision Tree Classifier	Without Stop word removal	0.489	0.489	0.489	0.656
	With stop word removal	0.499	0.499	0.499	0.665
	Lemmatization	0.559	0.559	0.559	0.553
	Stemming	0.57	0.57	0.567	0.575
Multinomial Naive Bayes	Without Stop word removal	0.582	0.582	0.594	0.5485
	With stop word removal	0.639	0.607	0.635	0.625
	Lemmatization	0.5985	0.5985	0.559	0.545
	Stemming	0.557	0.557	0.559	0.545

4 Discussions:

Sarcasm seems to be often accompanied by extra-linguistic cues in addition to linguistic or lexical indicators. Extra-linguistic cues used as features included intensifiers ("so," "too," "very") and abnormal capitalization (words with all letters capitalized or multiple words having the first letter capitalized).Thus a model that include extra-linguistic cues in the analysis of the corpus would be a good follow up. If annotated using sarcasm markers or self-labelled datasets, it is important to note that the labels may be applied incorrectly, often due to a lack of standardized methods. We suspect this to be the reason behind the model's failure to obtain a good accuracy/Micro-averaged precision value. Since Sarcasm is mainly dependent on context and word sense disambiguation CBOW, POS Tagging would have been ideal for the use case. Due to time and compute constraints, few of the techniques we were planning to implement could not be executed like Word embeddings and POS tagging. However, we plan on improving on these in the second phase of the project, while including Deep learning models. Possibly the largest ongoing issue with automatic sarcasm detection comes from stems from poor dataset annotation. Since we couldn't control for it, training the dataset on the models best known for word sense disambiguation and context imbibing attention mechanisms would hopefully work in our case.

5 References

- A survey on automated sarcasm detection on twitter

6 Contribution:

- Pipeline and Evaluation fitting

- Hyperparameter Tuning
- Report writing for Discussions, part of the methods and the Experiment Analysis