Citations Sentiment Analysis

Group: Priyadarshini Murugan, Rajal Nivargi
Computational Foundations of Informatics, Spring 2020
Professor Vasant G Honavar

May 1, 2020

Contents

1 Description

1.1 Background

A citation is a reference to a published or unpublished source.[6] The citation of a particular source is a combination of the in-text reference to the bibliographic entry of the source.There are various methods to cite an article in a alphanumeric entry. It indicates the position where the source is referenced in the text. Such a references helps easily identify the main idea the author is trying to draw from the cited text. Its main purpose is to acknowledge the relevance of the works of others to the topic in the text at the position where the citation appears.

The importance of citation is to protect intellectual property and avoid plagiarism by giving due credit to the course of the information. It also allows the reader understand if the claim made by an article is supported by previous work. It can help gauge the strength and credibility of the claim. As Roark and Emerson have argued, citations relate to the way authors perceive the substance of their work, their position in the academic system, and the moral equivalency of their place, substance, and words. [8].

A scientific citation is a reference made to a scientific publication which includes a book or article or technical papers, etc. These are related to the topic of discussion in the cited text. As said earlier, this assists the reader gain to access the new work, understand the background information and acknowledge the contribution of the authors. The citations may be regarding previous experimental procedures, source of data, ideas and theory, diagrams or results to compare, etc.

1.2 Introduction

As the amount of textual data is continuing increasing at an exponential rate on the World Wide Web, it is now become important to derive insights from this text. The opinion centric information retrieval is one of the important areas of interests. "Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes." [3] It is one of the most research prominent areas in Natural Language Processing with applications in understanding the brand image and product/service reviews. Opinion mining from Twitter data analysis, product reviews as well as movie reviews have been explored predominantly for marketing purposes.

With the availability of the scientific publications through digital libraries, accessing the text in these publications has become possible. The main motive of these digital libraries are to collect and store relevant papers at one place. Some examples of these are IEEE Xplore, Google Scholar, CiteSeer, etc. They allow researchers look for publication and citation information.

"Citation sentiment analysis is an application of sentiment analysis in citation content analysis, which aims to determine the sentiment polarity that the citation context carries towards the cited paper. In citation sentiment analysis, the opinion target is the cited paper, and the categories of sentiment polarity could be either positive, negative or neutral."[9] The extraction of the sentiments can be different from traditional opinion mining from reviews/tweets. The citations are made in a formal language and the use of strong sentiment is generally avoided. Hence, the usual negative words like 'bad', 'dislike', 'disagree', etc are not used. Instead, the same sentiment is expressed in a subtle way such as 'outerform', 'rather unexplored',etc. Thus, an algorithm to classify the sentiment expressed in the citation can be a challenging task.

1.3 Research questions

The aim of this project is to understand the sentiment towards a claim made by the cited paper in research papers.

The main question we would like to answer is: Is the classification of sentiment from scientific citations possible?
Given the difficulties with the expression of sentiment by scientific citations, we would like to know if an automatic analysis by an algorithm will be possible. There has been some past work in this area with attempts using different features and models. Hence, we are optimistic about the result.

What features will be important from the text?
Natural language processing is used to read, decipher and understand the human language in a manner that is valuable and interactable with a computer. An important step of this is extracting various texual features given

a sentence. We would like to know which of these said features will help us in this classification task.

Is it possible to improve the accuracy of existing methods?
The state of the art machine learning models are Support Vector Machine using Radial Basis Function kernel and Naive Bayes. Of these, SVM's are generally seen to perform better along with (1,3) n-grams. [10] We wanted to explore these models along with another method we will be suggesting at the end of the project and if our method could outperform the above.

1.4 Significance

Researchers typically need to analyze several scientific papers to find those relevant to their research. This task of analysis is cumbersome. An available sentiment analysis of such relevant papers can set the further research directions. This project also facilitates the estimation of the confidence of researchers in the claim made by the cited paper. This can also help ”recognise unaddressed issues and possible gaps in the current research, and thus help them set their research directions. Citation sentiment detection can also help researchers during search, by detecting problems with a particular approach”.[1] This can also help measure the impact of the article and to to improve the bibliometric measures. This can be a method of importance in the case of no ground truth. For example, in the current situation of COVID-19 being a completely unprecedented area of research, a study like this can help identify the papers which may be more relevant than the rest based on the sentiment/confidence of experts in the area citing the paper.

1.5 Project Objective

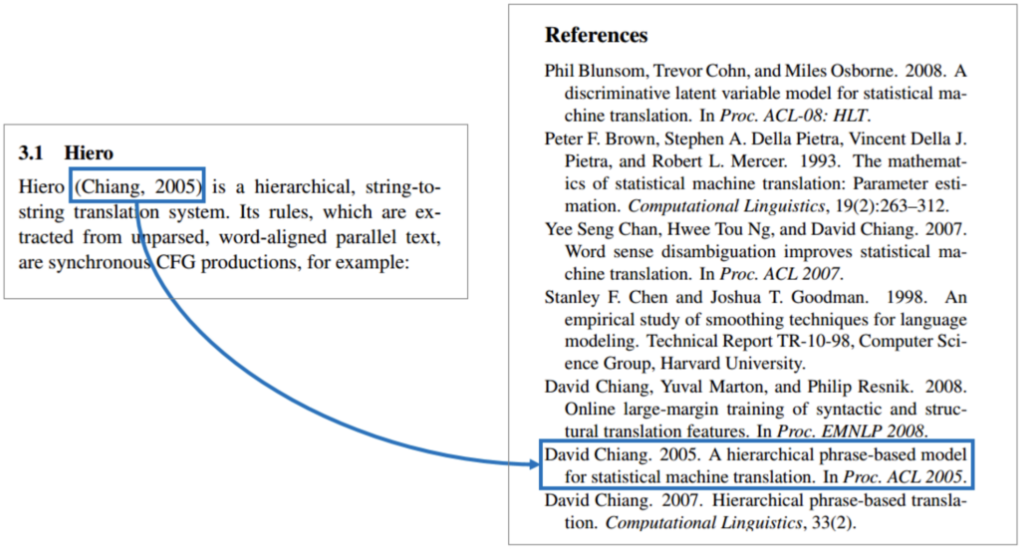
1. Find suitable data for the purpose of the project
2. Implement state of the art method
3. Identify other features that can be used to improve performance
4. Observe the results by using a different machine learning approach
5. Analysis of the observations and discuss if the last two steps improve the performance of existing methods

2 Data

The data used was obtained from the technical report by Awais Athar, 'Sentiment analysis of scientific citations'[1]. This dataset is made available for use by the author. The corpus consists of 8,736 citation sentences which have been manually annotated with sentiment. The sentiments are classified as:

- p : positive
- n : negative
- o : neutral

It has been extracted from ACL Anthology Network corpus.ACL Anthology digital library has papers and journals on Natural Language Processing and Computational Linguistics since 1965 in PDF format. Mostly the Harward style of citations is used in Computational Linguistics. Hence, the citations were machine readable. Citation sentences in the metadata contain the positional mapping to the sentence where the citation occurs. These were extracted to form this corpus with information of the source ID, target ID, sentiment and citation sentence.



3.1 Hiero

Hiero (Chiang, 2005) is a hierarchical, string-to-string translation system. Its rules, which are extracted from unparsed, word-aligned parallel text, are synchronous CFG productions, for example:

References

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proc. ACL-08: HLT*.

Peter F. Brown, Stephen A. Della Pietra, Vincent Della J. Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proc. ACL 2007*.

Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proc. EMNLP 2008*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. ACL 2005*.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2).

Figure 1: Example of citation and its refernce section entry [1]

3 Algorithmic abstraction

The process of sentiment analysis from the data involves transforming the text to measurable features a machine learning model understands. There are a variety of features that can be computed from given sentences. According to literature, we have narrowed our view to 5 of the features as explained in section 4.

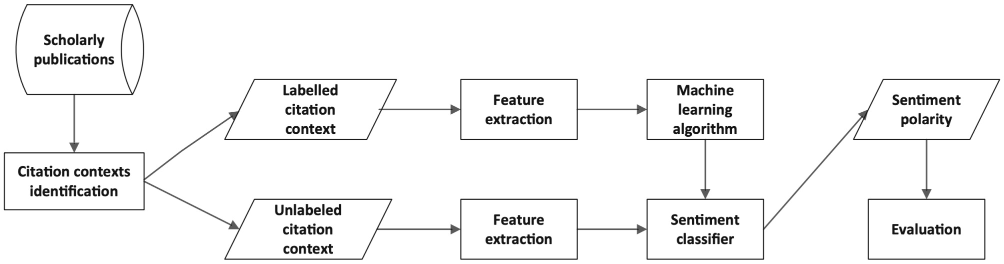

Scholarly publications
Citation contexts identification
Labelled citation context
Unlabeled citation context
Feature extraction
Feature extraction
Machine learning algorithm
Sentiment classifier
Sentiment polarity
Evaluation

Figure 2: Illustration of a sentiment analysis from scientific citations process[10]

The machine learning models that are generally used for this purpose are SVM and Naive bayes.[10] Support vector machines or SVM is a supervised machine learning algorithm which used a hyperplane to classify linear data. Naïve Bayes classifies using a family of simple ”probabilistic classifiers” based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. Along with the two, we propose using a neural network for this purpose. In this project we will be using a recurrent neural network architecture called Long Short Term Memory(LSTM). Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. It may be useful in understanding the dependence of the words in a citation sentence. Thus, we will also be using this deep learning model along with the two classification methods of SVM and Naive Bayes.

4 Features

4.0.1 N-grams

In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. According to literature, tri-grams with n=3 are shown to have better results than unigrams(n=1) and bigrams(n=2).

4.0.2 Part of Speech Tags

In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST) is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech. It is also called grammatical tagging or word-category disambiguation. It is useful in the processing of natural language because a words relative meaning can be understood by the model from its position with respect to adjacent and related words in a phrase, sentence, or paragraph which can help identify its definition and its context.

4.0.3 Lemmatization

For many languages, words appear in inflected forms. For example, in English, 'summary' can also be used as summarizing', 'summarization' or 'summarize'. They all belong to the same lemma of the word. This is closely related to stemming. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech. For example, 'better' had its lemma as 'good'. However, this is not detected by a stemmer.

Lemmatisation (or lemmatization) in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. In computational linguistics, lemmatisation is the algorithmic process of determining the lemma of a word based on its intended meaning.

4.0.4 Word2Vec

Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space.[7]

4.0.5 Lexicons

A lexicon, word-hoard, wordbook, or word-stock is the vocabulary of a person, language, or branch of knowledge (such as nautical or medical). In linguistics, a lexicon is a language's inventory of lexemes. In this case, we used the lexicons provided along with the corpus. It is specific to the widely used words in the scientific citations with 82 such words. These words are marked with polarity as -1 and 1 for negative and positive phrases respectively.

5 Implementation

The corpus we used contained 8736 labelled data of citations with source and cited paper ID. The dataset was already processed and cleaned by Author of the dataset. So, we did not need to do any further pre-processing for the dataset.

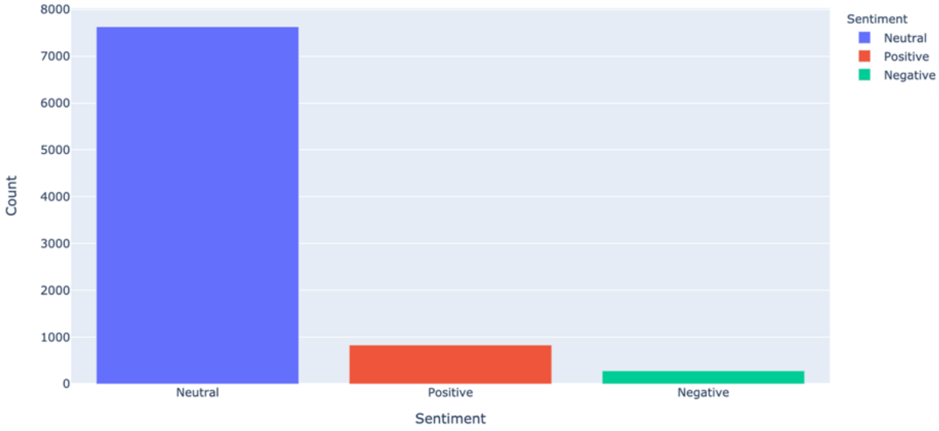


Figure 3: Sentiment Composition of the dataset

The figure 3 depicts the composition of labelled sentiments in the dataset. From this figure we can infer that most of citation text were labelled as neutral and that we had small composition of citation text labelled as Positive/Negative.

As any other classification problem, we have to train a classifier based on some feature of the sentiment classes. So, basically there are two sub-tasks:

1. Feature extraction process
2. Training the classifier

5.0.1 Tf-Idf weighted Word Count: Feature Extraction

Statistical approaches such as machine learning and deep learning work well with numerical data. However, natural language consists of words and sentences. Therefore, in order to build a sentiment analysis model, we need to convert text to numbers. Several approaches have been developed for converting text to numbers. N-grams, and Word2Vec model are some of them that we employed in our project.

TFIDF with n-grams converts textual data to numeric form, and is short for Term Frequency-Inverse Document Frequency. The vector value it yields is the product of two terms; TF and IDF. Relative term frequency is calculated for each term within each document as below.

$$TF(t,d) = \frac{\text{Number of times term(t) appears in document(d)}}{\text{Total number of terms in document(d)}} \tag{1}$$

Next, it gets Inverse Document Frequency, which measures how important a word is to differentiate each document by following the calculation as below.

$$IDF(t,D) = log(\frac{\text{Total number of documents(D)}}{Number of documents with term(t) in it}) \tag{2}$$

Once we have the values for TF and IDF, we can calculate TFIDF as below

$$TFIDF(t,d,D) = TF(t,D).IDF(t,D) \tag{3}$$

sklearn.`feature_extraction.text` is a library class implemented in sklearn library for extraction of text features. We extracted tf-idf weighted features with the help of its functions.
Upon initialization, `fit_transform`() function is called by vectorizer object with parameter `X_train` where `X_train` is a list(iterable) of strings representing content of the document.

`fit_transform(X_train)`does the following:

- Tokenizes words, preprocesses it for removing special characters, stop words etc. Also removes words that do not agree to the `token_pattern` regex.
- Creates a vocabulary of words with count in training set. Takes `max_features, min_df` and `max_df` in consideration.
- Finally, for each single string(document), it creates the tf-idf word count vector. The word count vector is a vector of all words in vocabulary with its frequency weighted by term frequency and inverse document frequency.

It returns a feature vectors matrix having a fixed length tf-idf weighted word count feature for each document in training set. This is also called term-document matrix.

5.1 Training and Evaluating the Text Classification Model

5.1.1 Train/Test Split

Before we train the model, we first had to split the data. We split the data into two chunks: train, test.

- Train set: The sample of data used for learning
- Test set: The sample of data used only to assess the performance of a final model.

The ratio used to split data was 80% of data as the training set, and 20% for the test set. We made sure that we have balanced distribution of sentiments in our train and test set. After splitting the data, we used the vectorizer obtained from previous to train our SVM and MultinomialNB classifiers and LSTM.

5.1.2 Long short-term memory(LSTM)

For LSTM, we used architecture depicted in Figure 5 for training the model. Steps were performed for LSTM's multi-class classification as follows.

1. Map words to word embeddings
2. Define LSTM model that receives a sequence of vectors as input and generates prediction.

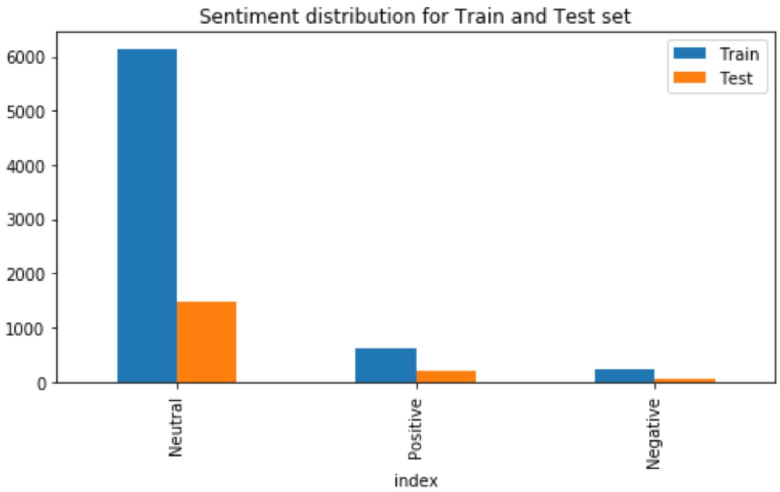


Figure 4: Sentiment distribution of train and test set

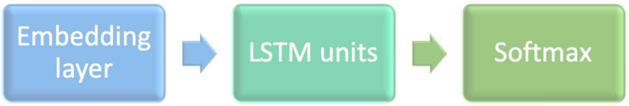


Figure 5: LSTM Architecture

we used Tokenizer word embedding model to learn the contextual relations among words in training data. This embedding model from Keras allows to vectorize a text corpus, by turning each text into either a sequence of integers (each integer being the index of a token in a dictionary) or into a vector where the coefficient for each token could be binary, based on word count, based on tf-idf.we set `num_words` in tokenizer to 2000.As our data size is comparatively smaller, we used this number. This means that the tokenizer detects the 2000 most frequent words from the dataset and use them as features for further model building.

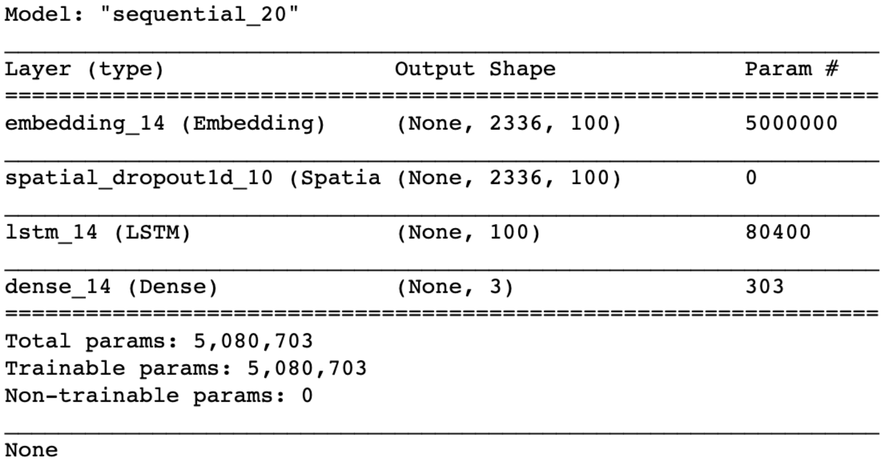
Then we initialized the sequence model and added the following layers for training the model

- Embedding layer: `embed_dim` as 100
- LSTM layer with 100 units

- Fully connected (dense) layer with 1 (output) neuron with the softmax activation function.

- `categorical_crossentropy` loss and Adam optimizer

Also, we set accuracy as the metric for measuring model's performance. You can see the summary of the model in the Figure 6.



```
Model: "sequential_20"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_14 (Embedding)     (None, 2336, 100)         5000000   
_________________________________________________________________
spatial_dropout1d_10 (Spatia (None, 2336, 100)         0         
_________________________________________________________________
lstm_14 (LSTM)               (None, 100)               80400     
_________________________________________________________________
dense_14 (Dense)             (None, 3)                 303       
=================================================================
Total params: 5,080,703
Trainable params: 5,080,703
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 6: Summary of LSTM model

5.1.3 Support Vector Machine

Support vector machine (SVM) is a learning technique that performs well on sentiment classification.This model was used to improve the accuracy obtained from LSTM. Linear kernel tend to perform better for sentiment classification than RBF kernel as the texts in general are considered to be linearly separable. We have tried using both in this project.

5.1.4 Naive Bayes

Multinomial Naive Bayes allows us to represent the features of the model as frequencies of their occurrences (how often some word is present in our review). In other words, it tells us that the probability distributions we're using are multinomial. we trained the model using the tf-idf vectorized train dataset.

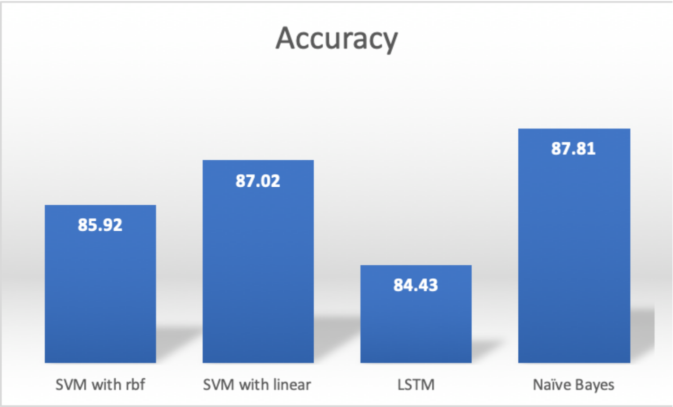


Figure 7: Accuracy

6 Results

The accuracy value represents the percentage of test texts which were classified correctly by the model. Results are shown in figure 7

In general, LSTM model does not seem to yield improvement in accuracy.It could be because of smaller data-set , LSTM tend to over fit the data and perform poorly on test set. We used for RBF kernel and linear kernel for prediction. The performance for linear kernel of 87.02% was slightly higher than RBF kernel of 8.92%. Compared to other models , Multi Nominal Naive Bayes provided better accuracy of 87.81. Consider Figure 8 for Confusion matrix of Multi Nominal Naive Bayes.It indicates that the model classified about 87.81 percentage of data correctly and misclassified about 12 percentage of the data incorrectly.

7 Conclusion

Through this project, we were able to understand the importance of citation and the opinion mining of the citations in the research community. It can help identify publications with a high confidence score from other researchers by using the sentiment towards its claim as a metric.
This project has also been a great first experience with using Natural Language Processing. The different features important in text analysis and machine learning where studied and implemented. Each of these features have their significance in terms of context, grammar and positioning. Thus, we were able to determine which features to use based on the data.

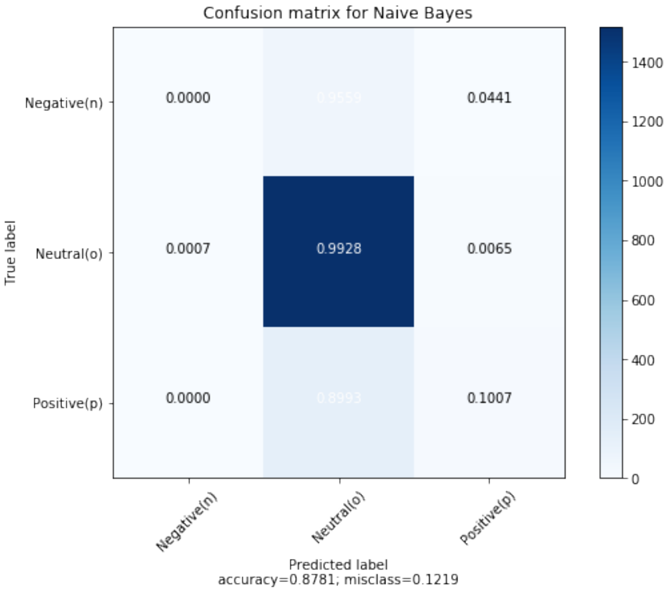


Figure 8: Confusion Matrix of MultiNominal Naive bayes

We tried three machine learning models and also integrating them with different textual features. Since this is a non-trivial task, some features did not work out as expected. Given the limited time, we could explore only some features and corresponding models. Further work would be attemping a combination of the features to be used as input to the models and observe the output. Since the difference between the models is not very high, it may be possible that some features work well with a particular model to yield a higher accuracy.

8 Group members

The two members in the group are Priyadarshini and Rajal Nivargi. Priyadarshini is a IST Master's graduate student and Rajal is a CSE Master's graduate student, both in the first year. The team members have experience in Data Science in social applications and hence, will be collaborating together.
Work distribution: We have been working together in the literature reviews by dividing the research papers and discussing the main points from each. The code as well has been a joint effort by understand the data and the functions to be used for the logic.

References

[1] A. Athar, "Sentiment analysis of scientific citations," in *Proceedings of the ACL 2011 Student Session*. Portland, OR, USA: Association for Computational Linguistics, June 2011, pp. 81–87. [Online]. Available: https://cl.awaisathar.com/citation-sentiment-corpus/

> In this paper, the approach to the problem uses classifier features such as n-grams, Part of Speech Tags and a science specific sentiment lexicon that is manually extracted from citations that are not the part of the data set. The classification algorithms used in this paper were Naive Bayes and Support Vector Machine that uses RBF(Radial Basis Function) kernel. It shows that the n-grams as trigrams and window based negation gives the best results for the SVM classifier on the basis of the F-score.

[2] C. T. Ghosh S., Das D., "Determining sentiment in citation text and analyzing its impact on the proposed ranking index," 2018.

> The main goal of this paper is to use the sentiment information in each citation instance(qualitative) in addition to the number of citations (quantitative) to determine the worth of the paper.They deployed a system of citation sentiment analysis to achieve three major objectives. First, to identify sentiments in the citation text and assign a score to each of the instances. They have used a supervised classifier for this purpose. Secondly, They have proposed a new index (called the M-index) which takes into account both the quantitative and qualitative factors while scoring a paper. Finally, They developed a ranking of research papers based on the M-index. This is done by extracting feature from corpus of Athar and groups to identify the sentiment polarity of the citation instances.They extracted features such as Automatic Sentiment,Positive polarity words, Negative polarity words,Presence of specific Part-Of-Speech tags,Presence of specific Dependency tags,Self Citation,Opinion Lexicons and then used the machine learning software WEKA2 to combine the above features to form a feature set and used the J48 classifier to generate a pruned C4.5 Decision Tree for three-way

classification of the citation instances – positive, negative and neutral.

[3] B. Liu, *Sentiment Analysis and Opinion Mining.* Morgan Claypool Publishers, 2012.

[4] H. Liu, "Sentiment analysis of citations using word2vec," 2017.

The author conducted empirical research to understand how word2vec works on the sentiment analysis of citations.The proposed method constructed sentence vectors (sent2vec) by averaging the word embeddings, which were learned from Anthology Collections (ACL-Embeddings)and polarity-specific word embeddings (PS-Embeddings) for classifying positive and negative citations. The sentence vectors formed a feature space, to which the examined citation sentence was mapped to. These features were input into classifiers (support vector machines) for supervised classification. Using 10-cross-validation scheme, evaluation was conducted on a set of annotated citations. The results showed that word embeddings are effective on classifying positive and negative citations.

[5] D. Mercier, A. Bhardwaj, A. Dengel, and S. Ahmed, "Senticite - an approach for publication sentiment analysis," *ArXiv*, vol. abs/1910.03498, 2018.

'SentiCite' shows an approach to sentiment analysis of citations. The datsets used were created using Scientific publications from International Conference on Document Analysis. The features used in this approach are obtained by Part of Speech tagger for determining type of token, length or capitalization. They use two types of classifiers: Support Vector Machine and a perceptron. Both the methods were facing problems of different natures. Hence, a multi-classifier approach based on both was shown to increase performance,

[6] Merriam-Webster, ""cite." merriam-webster.com dictionary," accessed 30 Apr. 2020. [Online]. Available: https://www.merriam-webster.com/dictionary/cite

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proceedings of Workshop at ICLR*, vol. 2013, 01 2013.

[8] M. L. Roark and W. Emerson, “Signals,” 2015. [Online]. Available: http://dx.doi.org/10.2139/ssrn.2688685

[9] W. Y. W. J. D. X. X. H. Xu J, Zhang Y, “Citation sentiment analysis in clinical trial papers.” American Medical Informatics Association, 2015.

> This study attempts to classify the sentiment polarity of a citation in clinical trial papers, i.e., the sentiment polarity relation between the citing and the cited paper, based on the sentiment analysis of the content of citation context. They first annotated a citation sentiment analysis corpus, which contains discussion sections extracted from 285 clinical trial papers. The citation sentiment polarity was annotated at the citation-level by following an annotation guideline. A simple rule-based method was used to extract the citation context, which is a set of on-topic sentences. The citation sentiment polarity was then classified using machine learning methods incorporating features extracted from the citation context. The performance of citation sentiment analysis was evaluated using the 10-fold cross-validation method

[10] A. Yousif, Z. Niu, J. Tarus, and A. Ahmad, “A survey on sentiment analysis of scientific citations,” *Artificial Intelligence Review*, 12 2017.

> Sentiment analysis of scientific citations is more complex and difficult than sentiment analysis of classical texts like comments, reviews, etc. The main challenge is that citations avoid the use of direct sentiments to be expressed. It is indirectly implied in a polite manner which makes it hard for simple sentiment analysis to be used in this context. There are also different types of citations i.e. Harvard, IEEE among other styles that can be used while extracting them from the technical papers. The citation sentences extracted require preprocessing in order to be used for further analysis. Among the machine learning methods that can be used for this purpose, SVM and Naive Bayes are the most widely used algorithms.Some other techniques include deep learning like LSTM with attention mechanism on all word vectors, lexicon based methods using adjectives determining polarity and keyword based methods.