ANALYSING TWITTER SENTIMENTS USING CONTEXT INFORMATION

A PROJECT DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by:

SHUBHANGI UPASANI	**2K15/EC/156**
VARNIKA GUPTA	**2K15/EC/177**

Under the supervision of:

DR. AKSHI KUMAR- DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROF. JEEBANANDA PANDA- DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042

MAY 2019

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042

CANDIDATES' DECLARATION

We, Shubhangi Upasani (2K15/EC/156) and Varnika Gupta (2K15/EC/177), students of B. Tech., Electronics and Communication Engineering, hereby declare that the project dissertation titled **"Analysing Twitter Sentiments using Context Information"** which is submitted by us to the Department of Electronics and Communication, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Bachelor in Technology, is original and not plagiarized from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Associateship, Fellowship, or any other similar title or recognition. The supervisor is the copyright holder of the work embodied in this report and no work can be published without taking consent from the supervisor.

Place: **New Delhi** Shubhangi Upasani | Varnika Gupta

Date: **28th May, 2019** 2K15/EC/156 | 2K15/EC/177

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042

CERTIFICATE

I hereby declare that the project dissertation titled **"Analysing Twitter Sentiments using Context Information"** which is submitted by Shubhangi Upasani (2K15/EC/156) and Varnika Gupta (2K15/EC/177), students of B. Tech. , Electronics and Communication Engineering, to the department of Electronics and Communication, Delhi Technological University, Delhi in partial fulfilment of the requirement of the award of the degree of Bachelors in Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: **New Delhi**

Date: **May 2019**

Dr. AKSHI KUMAR

SUPERVISOR

ASSISSTANT PROFESSOR, CSE DEPT.

PROF JEEBANANDA PANDA

SUPERVISOR

PROFESSOR, ECE DEPT.

ABSTRACT

Social media has become extremely popular in today's time as people often view it as a platform to voice their opinions and sentiments about various organisations, people, companies, products or events. Twitter is one such platform that gives its users an opportunity to express their views. The magnitude of opinion data and hence social intelligence on Twitter is continuously increasing with each passing day. There are endless possibilities to use this wealth of information for getting to know the popular sentiments of people and their mind-sets and attitudes. Organisations and firms use this data to connect to their customers better and widen their market reach. Government and political institutions use the same data to find out mass sentiment and ensure the welfare of their citizens. We have implemented Twitter sentiment analysis in this dissertation which is often not easy because opinions expressed on Twitter tend to belong to a wide range of domains. Another challenge is that tweets often have misspelt words and slangs which sometimes make the sentiment behind them obscure. Through this dissertation, we propose a hybrid approach for Twitter sentiment analysis. Our implementation makes use of both machine learning and lexicon based approaches and creates a hybrid model out of it. It also makes use of context which we define as semantic units larger than a single word. The aim is to get better sentiment polarities of tweets considering the context and thrive to see an improvement in the accuracy of sentiment analysis.

ACKNOWLEDGEMENT

The success of this dissertation required guidance and support from many people and we take this opportunity to express our sincere gratitude to all of them.

First and foremost, we would like to thank our mentors, Dr. Akshi Kumar, Department of Computer Science and Engineering, Delhi Technological University and Prof. Jeebananda Panda, Department of Electronics and Communication, Delhi Technological University for their immensely valuable guidance and constant supervision. Dr. Akshi Kumar's constructive advice and support made this experience truly enriching for us and helped us to streamline our efforts in the right direction. Her objectivity and meticulousness is something we wish to emulate. Prof. Jeebananda Panda's acuity and untiring enthusiasm was inspirational for us and constantly motivated us to thrive for more. His selfless and sympathetic nature cannot be described adequately by mere words.

We would like to thank, Geetanjali Garg, Department of Computer Science and Engineering, Delhi Technological University, who was always there for us in times of need and provided us with the necessary technical knowledge in completion of the dissertation.

Lastly, we would also like to extend our thanks to Prof. S. Indu, H.O.D, Department of Electronics and Communication, Delhi Technological University for her unwavering support. She encouraged us to pursue our interests and undertake an inter-departmental major project. We owe our deepest gratitude to her.

SHUBHANGI UPASANI VARNIKA GUPTA

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODCUTION TO SENTIMENT ANALYSIS

With the growth of Internet, social media and micro blogging platforms like Facebook, Twitter, Tumblr have come to dominate news and trending topics around the globe at a rapid pace. A topic tends to become trending if more and more people express their opinions about it, thus making it a source of online perception. These topics are more than often related to consumerism, politics, governments, enterprises and organisations. Large firms and organisations tend to take advantage of these opinions to improve their products and services and hence their marketing strategies. There is a huge potential to discover fascinating consumer behavioural patterns from the infinite social data available for business-driven applications.

Sentiment Analysis, also called opinion mining, is the field of study that analyses people's opinions, sentiments, evaluations, appraisals, attitudes and emotions towards entities such as products, services, organisations, individuals, issues, events, topics and their attributes. These opinions are usually obtained in the form of reviews. There are several tasks related to this problem space known by various names like *sentiment analysis, opinion mining, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, review mining etc.* Sentiment analysis takes these opinions and classifies them as positive, negative or neutral. This field has gained fresh momentum as a research area after 2000's because of its wide range of applications in varied domains.

The main reason why sentiment analysis has become so popular is because of proliferation of commercial advertisements. The quantum of opinionated data on social media is so vast that it now finds use in tracking customer reactions, monitoring competitions, anticipating election outcomes and predicting investment trends and box office revenues. For example, many people these days use social networking sites for networking with other people and for staying updated on news and current evets. These sites like Facebook, Google+, Instagram and Twitter offer people a platform to voice their opinions. A person might buy a product and quickly post a review about it on social media stating its various pros and cons. This can form the basis of knowledge of this product for others and affect their decision to buy it. It is therefore critical to exploit this social intelligence to understand the reason behind certain sentiments and comments and use this information for not only marketing but other kinds of social studies as well. It is safe to say that research in sentiment analysis has not only had a profound impact in the field of natural language processing but has also been vital to management sciences, political sciences, economics and social sciences as they are all affected by people's opinions.

In general, sentiment analysis is defined at three levels:

- **Document level:** Sentiment analysis at the document level usually classifies whether the entire document conveys a positive or negative sentiment. For example, document level sentiment analysis tries to classify the sentiment expressed in a given product review as positive or negative sentiment. This task is also popularly known as document-level sentiment classification.

- **Sentence level:** The task is concerned with finding the sentiment of each sentence and classifying it into positive, negative or neutral. Neutral indicates that the opinion holder has no particular sentiment towards the object. This is also called subjectivity classification which differentiates objective sentences that convey some facts from subjective sentences that convey some opinion or sentiment.

- **Entity and Aspect level:** Aspect-level sentiment analysis is a more fine-grained kind of analysis. This was earlier known as feature-based opinion mining and summarization. Opinions are usually expressed in relation to a target. For example, the sentence, *although the colour is not that great, I still love the car*, has a positive sentiment about the car but negative sentiment about the colour. Thus, this sentence has a positive tone to it but can't be categorised as positive entirely. Aspect-level analysis hence discovers sentiment on entities (car) and their related aspects (colour).

Twitter sentiment analysis has become popular recently due its wide range of applications in commercial and management sectors. Twitter is an online networking site driven by tweets that have a maximum length of 140 characters. As of now, 65,000 tweets are published every second with an aggregate of 561.6 million tweets per day. These tweets are generally about a plethora of topics and are usually in an unstructured and unfiltered format. Twitter sentiment analysis is the process of analysing underlying sentiment in tweets. We aim to perform the same through this dissertation.

1.1 Background study on Twitter Sentiment Analysis

Sentiment analysis has become a popular research area in the past few years. We conclude the related work that has been done in the field of Twitter Sentiment Analysis below:

- E Junqué de Fortuny, T De Smedt, D Martens 2010 [14]

They proposed a method for sentiment analysis with their subject of study being the Belgian elections of 2010. A web crawler and a pattern mining module written in python was used. The module consisted of 3000 Dutch sentiment adjectives which were given polarities manually. The paper focussed on finding mentions of political parties and polarity counts of adjectives were calculated in a window surrounding the name of the party. The window size was two sentences long, both before and after the name.

- L Chen, W Wang, M Nagarajan, S Wang, AP Sheth - ICWSM, 2012 [13]

The paper presented a novel approach for drawing out sentiment expressions for a given target. The sentiment polarity is examined first and a set of words are obtained. The sentiment words help in identifying the target word followed by consistency and inconsistency relations.

- Johan Bollen, Huina Mao, Alberto Pepe (2011) [11]

They worked on global mood detection. Their research focussed on finding a connection between major political, cultural, socioeconomic or natural events to the widespread state of mind of people through tweets published on the same day. They used a scoring technique that counts the number of adjectives for all possible states of mind.

- A Tumasjan, TO Sprenger and PG Sandner (2010) [12]

This paper investigated whether Twitter is used as a platform for political discussion or does it just mirror offline political sentiment. The context of German Federal elections was used for the same. They found out that tweets can be considered as quite an accurate estimate of vote share.

- B O'Connor, R Balasubramanyan, BR Routledge 2010 [10]

They intend to analyse the publicly available data to infer population attitudes. A correlation of 80% was found out although the results varied greatly over the dataset. They used Consumer Confidence to get to know people’s opinions in public polls. Consumer Confidence is the measure of how consumers feel collectively about the prosperity of the economy.

1.2 Sentiment Analysis Applications

Over the years, sentiment analysis has found several applications in a wide range of domains, from consumer products and services, healthcare and finance to social events and political elections. Some of the applications are listed below:

- **Voice of customer:**

Sentiment analysis of social media reviews, mentions and surveys help to convey the voice of customers to consumer products and services companies. This way the companies get to know how common people feel about their products. This helps companies expand their markets and build loyalty among their customers.

- **Individual decision-making**

Nowadays, for buying consumer products and services, people no longer only rely on asking their friends or families for reviews and opinions on the same. There are many public forums where users can post reviews and discuss about various products and services which help other consumers take careful decisions about buying them.

- **Politics**

Voters usually use social media to get to know the popular mindset about a political candidate before taking a voting decision during elections. Opinionated postings on social media have impacted our social and political systems greatly. Such postings often mobilise masses for some political or social change such as those that happened during the Arab Spring of 2011.

- **Search Engine Optimization**

Opinion mining helps in discovering hot search keywords. Finding such keywords help brands in SEO (Search Engine Optimization). Opinion mining helps these brands come up with novel strategies about how their brand names can come up among the top results when trending or hot keywords are searched in a search engine.

- **Employee feedback:**

Sentiment analysis plays a huge role in getting accurate feedback from company employees and assessing their attitudes towards their jobs. This also helps in determining the level of employee satisfaction.

- **Better services**

Sentiment Analysis helps companies in determining which products or services of theirs are receiving the most negative reviews from customers. This way the company can get to know what is not working for them and where the problem is arising and subsequently they can rectify these problems.

- **Get to know what's trending**

This helps companies in staying updated and connecting with a wider audience base. This also bolsters the development of new ideas for developing products. The companies get to know the audience's demands and develops products accordingly.

1.3 Sentiment Analysis Research

Sentiment analysis is a popular research problem because of the full gamut of real-life applications it finds. It is a demanding Natural Language processing (NLP) research topic. The research in computational linguistics was limited before 2000's because there was not much opinion text found in digital format. Since then, this field has gained rapid momentum and is now regarded as one of the most engaging areas of research in NLP. Some applications of it include data mining, web mining and information retrieval. With the surge in its popularity, its applications have now expanded from computer science to management sciences.

1.4 Objective and Motivation

The magnitude of data pertaining to people's opinions on Web, particularly on social media platforms is continuously increasing and classifying these opinions according to their polarities is important for providing useful insights into the popular sentiment surrounding different entities in varied domains. Sentiment analysis has tremendous hidden potential and the demand for accurate techniques for capturing sentiment is only going to rise progressively. Therefore, we decided to embark on this journey to build a framework that analyses a solution for sentiment analysis and sentiment classification at the very fine-grained level- namely the sentence level and try to improve the accuracy of the same.

Moreover, the applicability of accurate sentiment analysis is quickly expanding by leaps and bounds, affecting decision-making in various inter-dependent and independent spheres like businesses, e-commerce, government, politics etc. Therefore, an improvement in sentiment analysis techniques is going to have a widespread impact, echoing through the above-mentioned domains and this is the biggest motivator behind us in venturing on this endeavour.

CHAPTER 2

SENTIMENT ANALYSIS METHODOLOGY

.

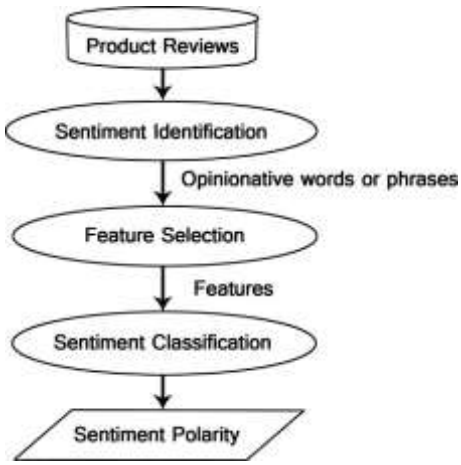


Figure 2.1 Sentiment analysis framework for reviews of various products

Figure 2.1 shows the most widely adopted framework for sentiment analysis on product reviews. For this particular case, the sentiment is identified by looking for opinionative phrases like '*a really bad customer service*' or words like '*good*', '*happy*', '*excellent*', '*bad*', '*horrible*' etc. Appropriate features are selected from these reviews during the feature selection process. There are broadly two kinds of features: explicit and implicit. After feature extraction, the resultant features are fed into sentiment classifiers to get the polarity of sentiment (positive, negative or neutral). This can also be used as an implementation strategy for other kinds of sentiment analysis.

The following sections describe the methodology of sentiment analysis in detail:

2.1. Basic Concepts

Given below is some general terminology associated with sentiment analysis.

- **Opinion**- It is any subjective expression that describes the emotions and sentiments of a person. Opinions also include performance assessment about an object, entity or event and their characteristics.

- **Object**- Any real world entity can be described as an object. An object could be a person, service, event, organization, topic etc. Every object has some attributes. For example, 'phone' is an object and some of its attributes are 'battery', 'speaker', 'screen', 'quality of voice' etc.

- **Opinion passage-** It is a collection of opinion phrases about various features of an object expressed in a document. These opinions can be positive or negative ones and usually pertain to some feature of the object. For example the sentence, *The picture quality of the camera is good but the batter life is terrible* conveys a positive and negative opinion on the features 'picture quality' and 'battery life' of the object 'camera'.

- **Features-** Users express their opinions on particular features of an object. These opinions have to be segregated accurately based on whether they are about explicit or implicit features of an object. A feature is called explicit if it or it's alternatives are present in the review sentences. For example, *the colour of the fruit is pleasant* has an explicit feature-'colour'. An implied or indirect feature is called an implicit feature. For example, in the sentence, *The fruit is big,* an opinion about the size of the fruit is conveyed.

- **Opinion holders or opinion sources-** Opinions on objects are expressed by users. The users are authors of opinions. Such users are called holders of opinions or opinion sources.

2.2 Feature selection

Sentiment analysis task is considered a sentiment classification problem. The first step in the sentiment classification problem is to extract and select appropriate text features. Features most popularly used are:

- **Term's presence and frequency**: The terms can be single words or n-grams (i.e. sequence of n terms). This method gives features either a binary value (one if the word is present in the document or zero otherwise) or uses weights to indicate their relative significance.

- **Parts of speech (POS)**: It is the practice involving reading sentences and assigning a POS tag (noun, adjective, verb etc) to each word of the sentence. This can be used for finding adjectives which are important since they indicate opinions.

- **Opinion words and phrases**: These include words and phrases that convey some opinions including '*good*', '*bad*', '*like*' or '*hate*'. Sometimes, text use phrases that express opinions without using any explicit opinion words. For example: the sentence '*cost me an arm and a leg*' expresses an opinion without using any explicit phrases.

- **Negations**: negation words are also important for extracting underlying sentiments in the text like '*not good*' is equivalent to '*bad*'.

Feature selection methods: These methods can be divided into **(i) lexicon-based methods** that need manual annotation and **(ii) statistical methods** which work

automatically. Lexicon-based approaches need a small collection of words usually called 'seed' words. A larger lexicon is obtained through bootstrapping the initial seed through synonym detection using WordNet or any other online resources. Statistical approaches are usually automatic on the contrary.

The feature selected from the text are either a group of words known as Bag of Words (BOWs) or a string of n-terms called n-grams. N-grams retain the order of words in the document. BOW is more simple and easy to use and therefore is the preferred methods for classification process. The feature selection is usually done after removing stop words and stemming or lemmatization (returning the word to its stem or root i.e. flies→ fly).

Current Problems in Feature Selection: A very challenging task is to extract features that detect irony. The aim is to identify ironical text or phrases like '*ISIS execution continues with a smile*'. Sarcastic reviews also pose a problem when it comes to their detection. For example, *What an awesome phone!! It crashed in two days* would be categorised as a positive review in most cases. Many sentences without opinion words can also express an opinion, like *This phone uses a lot of power to charge up* conveys a negative opinion about the phone. But such kinds of opinions are often missed in feature selection.

2.3 Sentiment Classification Techniques

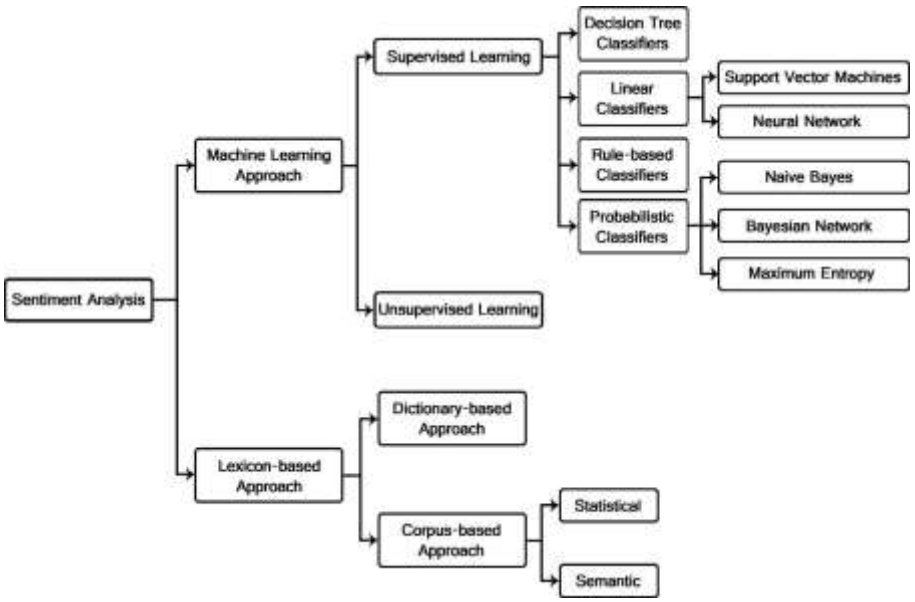


Figure 2.3.1 Sentiment Classification

Existing sentiment classification techniques are grouped into two main categories:

- **Machine learning approach:** This approach develops a model for the training the classifier using labelled examples or what we call formally as the training data. This means that all classifiers have a training phase first where they get to see several labelled examples, in this case labelled tweets as positive or negative. After training, these classifiers run on the test data, i.e. unseen data and classify that into one of the two classes described above.

  Machine learning methods can be **supervised** or **unsupervised**. Supervised methods need large amount of labelled data while unsupervised approaches work with unlabelled data. For text classification purposes, we generally work with supervised methods.

- **Lexicon-based approach:** This approach uses dictionaries of words previously annotated with semantic polarities and sentiment strengths. These dictionaries then calculate the overall score for the document. Some examples of the most common dictionaries or lexicons used are SentiWordNet, MPQA, Thelwall etc. There are two methods pertaining

to this approach: The **dictionary-based approach** which finds seed words first and then finds their synonyms and antonyms and the **corpus-based approach** which begins with a seed list of opinion words and then finds other words from a larger corpus. This is usually achieved using statistical or sematic methods. Each method is explained in more detail below:

I. **Dictionary based approach-** A small collection of words with known polarities- positive or negative is developed. This small set of words is then expanded using WordNet or any other online dictionary available that have synonyms and antonyms. The new words are then added to the previous collection before repeating the entire process. We stop iterating when no new words are found. A manual checking is needed to remove errors or ambiguities.

II. **Corpus-based approach-** This method finds opinion words with precise semantic orientations. We start with a small collection of opinion words and then expand this list using a domain-specific corpus. This is done by finding specific patterns that occur together with our opinion words. This approach has been tried in two settings (i) the one described above, given a set of seed words, find more sentiment polarity words from the domain corpus and (ii) adapt a general purpose lexicon to a new one using a domain corpus.

2.4 Fields related to Sentiment Analysis

There are various fields of study that work under the umbrella of sentiment analysis and have become areas of research recently. These are as follows:

- **Emotion detection-** Sentiment Analysis discovers opinions about entities. Due to the prevalent ambiguity between sentiment, opinion and emotion, researchers have defined emotion detection as a transitional concept that sheds light on people's attitudes towards an entity. The difference between sentiment and emotion is that sentiment reflects feelings while emotion describes attitudes.

- **Building resources-** This task concerns itself with creating lexica, dictionaries and corpora in which opinion expressions are classified according to their polarity. Building resources is not a sentiment analysis task per se but it does help in improving the accuracy of sentiment analysis and emotion detection. The main challenges that this task faces are ambiguity of words, multilingualism, granularity and the difference in opinion expressions among textual genres.

- **Transfer learning-** It draws parallels from additional sources to enhance learning in the concerned field of study. It is primarily used to enhance the accuracy of many text mining tasks like text classification, sentiment analysis, named entity recognition and part-of-speech tagging. Transfer learning in sentiment analysis can be employed for sentiment classification from one domain to another building a bridge between two domains.

CHAPTER 3

CONTEXT IN SENTIMENT ANALYSIS

Analysing context plays a huge role in sentiment analysis. Words often change their polarity with the context they are used in. For example, the word '*good*' conveys a different sense in each of the following sentences: *She gave him some pretty good insults!!* ('good insults' actually mean 'very bad insults'), *He has always fought the good fight against oppression* ('good fight' means 'trying very hard to do what is right') and *The restaurant was not good at all* ('not good' actually means 'bad' here). From the above examples we can conclude that taking context of a word into consideration is extremely important for correctly classifying its polarity.

The use of context to understand the meaning of the word is formally called contextual semantics. It simply means that we are discussing the overall meaning of the words in our document based on how they are used together. Contextual analysis helps to examine text in its social, cultural or historical context. Natural language processing and information retrieval are some areas that make use of contextual semantics, also known as statistical semantics. Contextual analysis has now come to be described as a method of studying a document by finding the words that appear in the document and analysing their relationships between one another to disambiguate their meanings and provide a comprehensive contextual understanding of what has been written.

There are several ways of capturing the context. Some of them have been described below:

- **N-grams**: n-grams help in considering the neighbouring words of a particular target word to understand the context in which the target word has been used.

- **Inter-sentential-** This method of capturing context takes into account few sentences in the neighbourhood of the target word whose meaning needs to be determined. The number and pattern of taking sentences are predetermined.

- **Shifter words:** Shifter words usually indicate a change in the tone of the sentence. For example, '*The restaurant is good but .....*' indicates that something negative about the restaurant is about to come. The word 'but' changes the tone of the sentence from positive to negative.

- **Co-occurrence:** This approach first finds words that co-occur (i.e. occur together) with a particular word and then compute it's contextual semantics. These words do not have to be adjacent. The underlying idea is that two words co-occurring tend to have similar meanings. Co-occurrence is a measure that indicates proximity between two entities in the semantic sense.

- **Ontologies:** Entities are first extracted from the tweets (example 'ISIS', 'Syria', 'United Nations') and then elaborated through their related semantic groups (like 'Jihadist group', 'country', 'organisation') using ontologies. They are helpful in introducing relationships between words.

- **WordNet:** It measures the sematic relatedness between different words and concepts by combining gloss information with sematic relationships. The gloss information is usually derived from synonym sets of words (called synsets) and the relationships between words in the same synset and those in different synsets is computed.

For the purpose of finding contextual semantics in our project, we find co-occurrence patterns between words to get better insights into the word polarity at a micro level and sentence polarity at a macro level. The technique adopted for finding co-occurrence patterns between words has been described in the following sections.

CHAPTER 4

PROPOSED METHODOLOGY

As discussed in Section 2.3, sentiment analysis is implemented using two broad approaches: machine learning approach and lexicon based approach. A comparative study between these two methods was accomplished first, the results of which are shown below:

Machine Learning approach	Lexicon based approach
It requires a good amount of labelled data for the training phase of classifiers. It is often the case that such labelled data is not available easily and a huge amount of manual labour and effort has to be expended in labelling the available data. This therefore is a labour-intensive approach.	It, on the other hand has no such stringent data requirements. It has the ability to grow the lexicon itself using online lexical resources (like WordNet) if given an appropriate initial seed list of words. Although the seed list has to be annotated manually, it is very easy and requires little time.
There is a large amount of unwanted information in the data we use which is referred to as noise. The machine learning algorithms we use are not able to differentiate noise.	Lexicons are not tailored to noisy data since they only have a fixed number of words in them. Hence, they differentiate between useful and noisy data better.
This approach is usually domain independent. It doesn't depend on the domain we are working on. Therefore, the approach can be implemented on data from	The lexicon based approach is domain-specific and needs to be retrained every time the domain changes. The transition from one domain to another is not as easy

different domains very easily.	as machine learning.
There is no way to take context of words into account. Machine learning approaches would treat every word objectively. As a result, for a task like sentiment analysis, the accuracy tends to be limited.	With lexicons, we can take context of a word into consideration. Using context with lexicon gives a better classification of the sentiment.

Table 4.1 Comparison between Machine Learning and Lexicon-based approaches

Comparing the pros and cons of the two approaches, we decided to adopt both of them for their individual merits and hence build a hybrid model out of it. Since our Twitter data is not domain specific (it has multi topic data- from healthcare to politics to music etc.), we needed to use machine learning for its domain independence. We use several machine learning classifiers for sentiment classification of our tweets into positive and negative polarities. However, for a more accurate sentiment analysis, we wanted to correctly identifying the sentiment polarity of each word using its context. Therefore, we decided to use the lexicon-based approach as well. The context is extracted by finding co-occurrence patterns of words using a lexicon. To get the contextual semantics of a target word, we first find all the words that co-occur with our target word. Then we find the sentiment polarities of all these co-occurring words using the lexicon VADER. This lexicon returns positive, neutral and negative score of each word along with the compound score to indicate its overall polarity. We find sentiment of the target word by considering its co-occurring words i.e. the words that occur with it and their polarities. We do this for each word in a single document i.e. a single tweet to get the overall polarity of the tweet. The two approaches i.e. machine learning and context based approaches are then aggregated to obtain the final results. We aim to increase the number of classes from two (positive and negative) as originally present in the dataset to five (very positive, positive, neutral, negative and very negative). The below diagram shows the system architecture we have followed:

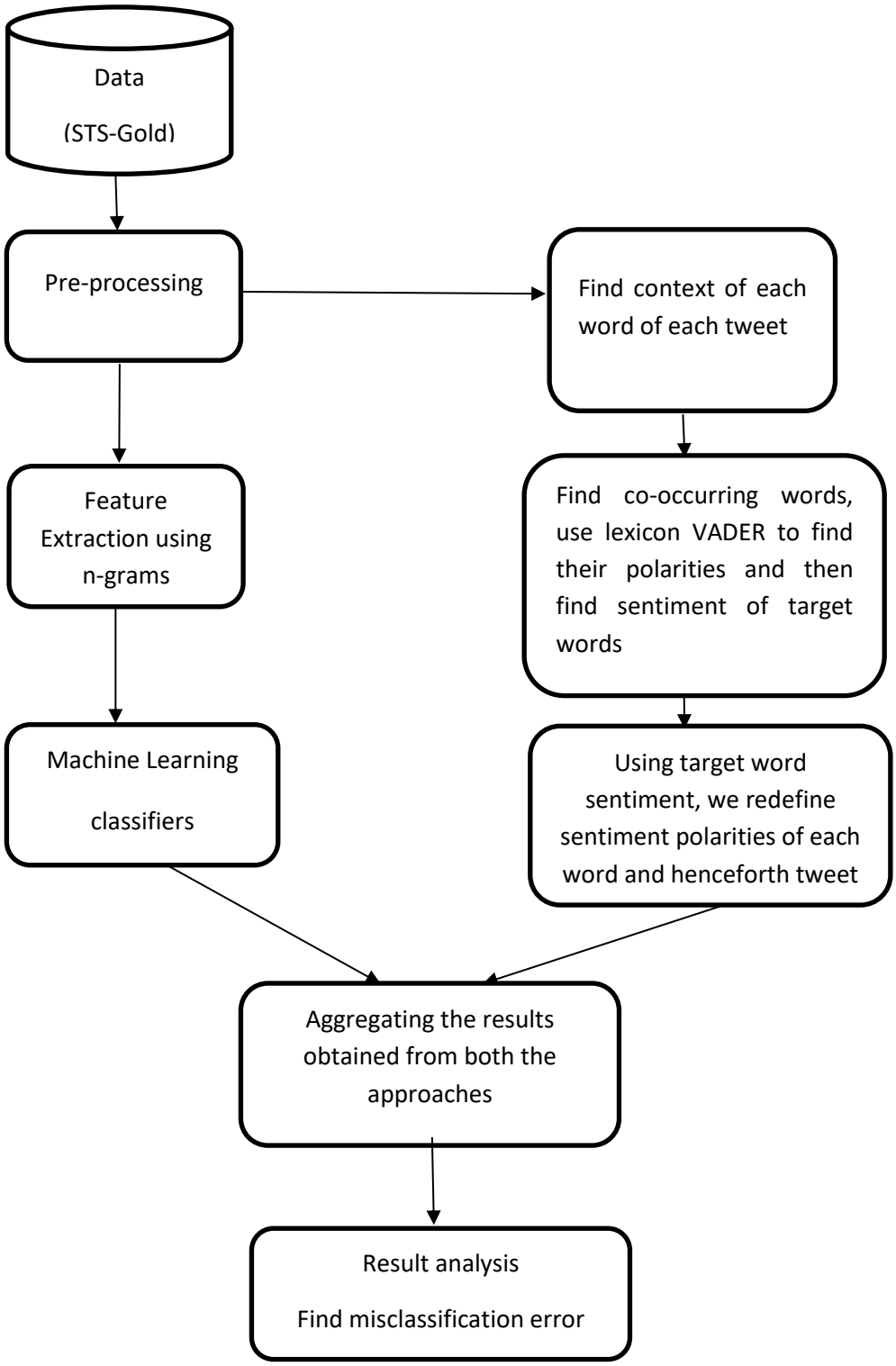


Figure 4.1 System Architecture

Our implementation has been explained in detail below:

4.1. Dataset and Pre processing

The dataset used is the STS-Gold dataset which has a total of 2,304 tweets. The dataset has tweets related to 28 different entities and has 1402 negative tweets and 632 positive tweets. A snapshot of the dataset is included in the Appendix 1. The dataset was made with the aim of complementing existing Twitter sentiment analysis evaluation datasets by annotating tweets and entities independently, allowing for different sentiment labels. This allows for both sentiment analysis of tweets and entities. The datasets that have been released till date hardly address the problem of entity-based sentiment analysis. The STS-Gold dataset supports the performance assessment of entity-based sentiment analysis models.

The STS-Gold dataset has the following columns:

- id- the tweet id (each tweet has a unique id)
- polarity- the sentiment polarity of the tweet (0:negative, 4:positive- STS dataset gives only two polarities to each tweet)
- tweet-the main text of the tweet

The data pre-processing often affects the execution of supervised learning algorithms. The broad steps we implemented for pre-processing of our data are as follows-

- **Case Conversion:** Words are made case-independent i.e. all words are converted into lower case in order to remove the difference between same words in different cases like “Text” and “text”.

- **Removal of hashtags, tagged words and emoticons:** All the tagged words starting with "@" and emoticons which are represented as "&amps" and "&quots" in the tweets are removed as they do not contribute towards analysing the sentiment. Also hashtags (#) are filtered out.

- **Punctuation Removal:** The data is ridden of all punctuation marks as they bear no significance for language analysis. Therefore, they are filtered out during pre-processing.

- **Removal of urls**: Urls starting with "*http*" and "*www*" are removed. Since urls do not contribute to the sentiment of our tweets, we remove them while pre-processing the dataset.

- **Stop-words Removal:** Stop words are the commonly used words like a, an, the, has, have etc. They are filtered out because they are unnecessary and do not contribute towards sentiment analysis. Our stop word list is prepared manually. We purposefully excluded the stop words like not, doesn't, don't, shouldn't from our list since they usually indicate a negative sentiment orientation. We retain these words and use them with their immediate adjacent neighbouring word to form a two-word phrase. This phrase is then analysed just like the rest of the words are.

- **Lemmatisation:** Lemmatisation in linguistics is the process of replacing the inflected forms of words by a single root form or lemma so they can be analysed as a single item. For example, the words 'sing', 'sang' and 'sung' are all converted to the same lemma 'sing' during lemmatization.

- **Spelling correction:** All spellings are corrected using Textblob library in python. Words like "luv" and "loveee" having the same connotation are corrected to the word "love".

4.2. Feature extraction

After pre-processing of data, features are extracted using n-grams method, particularly using unigrams, bigrams, trigrams and quadrigrams. In computational linguistics, N-grams are a running sequences of n items from a given sample text or speech. The constituents of n-grams are phonemes, syllables, letters, words or base pairs. An *n*-gram of size 1 is referred to as a "unigram"; size 2 is a “bigram”; size 3 is a “trigram” and so on.

Our dataset is divided into the training and testing data. All possible n-grams are first extracted from the pre-processed data, more specifically the pre-processed training data. These extracted n-grams form our features. These features together make the feature set. Then each tweet in the training set is analysed to mark which features are present in the tweet. The presence of each feature in the feature set is marked by a ‘1’ and the absence by a ‘0’. Therefore, after the feature extraction step, each tweet now transforms into a feature vector like [1, 1, 0, 0, 0, 1, 0, ……] where the length of the vector is equal to the total number of features in the feature set. On combining the feature vectors of each tweet, we get a feature matrix of dimension M x N where M is the total number of tweets in the training data and N is the total number of features. The matrix looks something like:

	Feature 1	Feature 2	Feature 3					Feature N
Tweet 1	1	0	1	...	...	...	...	1
Tweet 2	0	1	1	...	...	...	...	0
Tweet 3	1	0	1					0
	...	...	...					...
	...	...	...					...
	...	...	...					...
	...	...	...					...
	...	...	...					...
	...	...	...					...
Tweet M	0	1	1	...	...	...	...	0

Figure 4.2.1 Feature Matrix

The feature matrix is then processed via machine learning which has been described next.

4.3. Machine Learning Approach

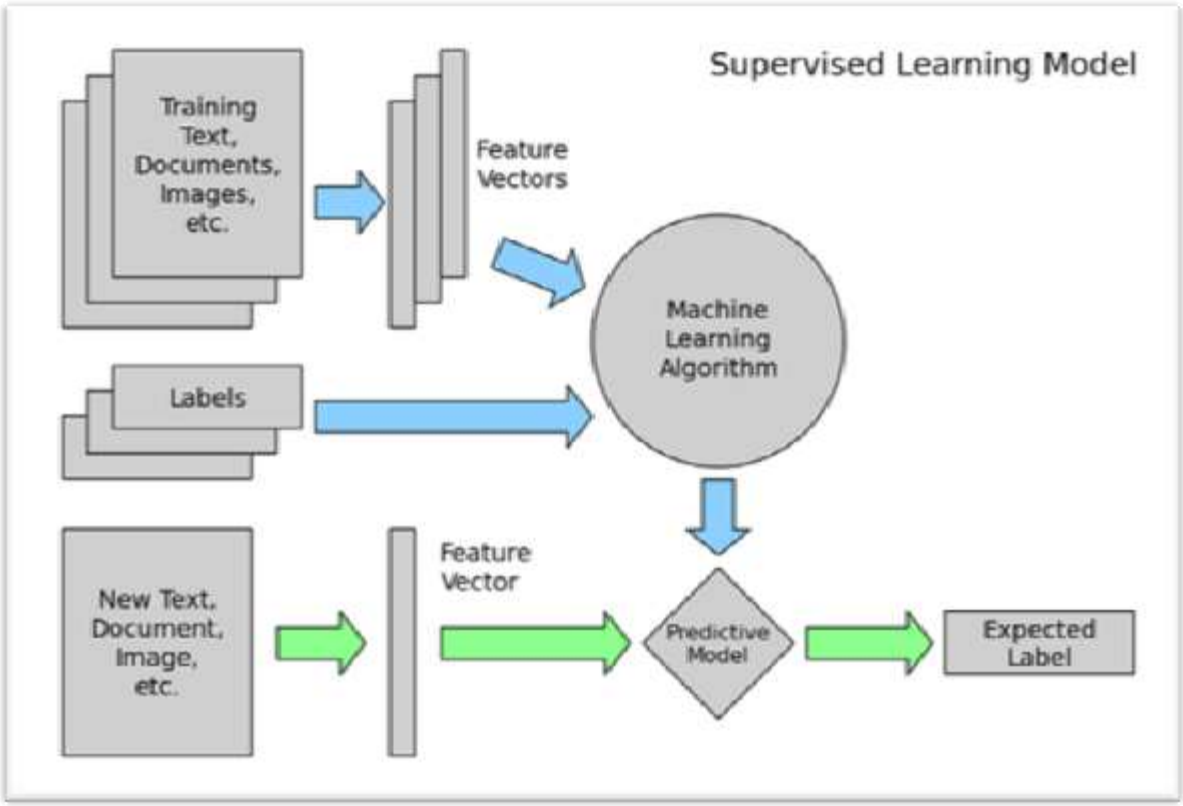


Figure 4.3.1 Supervised learning model

Supervised learning classifiers have been used for the purpose of sentiment classification. The dataset is first divided into training and testing data. The training data has the labelled data. The feature matrix obtained earlier from the training data is combined with the corresponding labels. This forms the data for training our machine learning classifiers.

	Feature 1	Feature 2	Feature 3					Feature N	Label
Tweet 1	1	0	1	...	...	...	...	1	0
Tweet 2	0	1	1	...	...	...	...	0	4
Tweet 3	1	0	1					0	4
	...	...	...					...	...
	...	...	...					...	...
	...	...	...					...	...
	...	...	...					...	...
	...	...	...					...	...
	...	...	...					...	...
Tweet M	0	1	1	...	...	...	...	0	0

Figure 4.3.2 Training data

The label 0 stands for a negative tweet and 4 imply that the tweet is positive.

The testing data, on the other hand has no labels. The labels for this data are predicted by the machine learning classifiers. The predicted labels are denoted by **MLScores** in our implementation.

The labelled training data is fed to several machine learning classifiers that have been described below. We use four major classifiers:

- **Support Vector Machine (SVM's)** – The idea behind SVMs is to find out linear separators or hyperplanes to separate different classes in the search space effectively. There can be several arrangements of hyperplanes that separate classes but the one that is chosen is such that the normal distance of any data point from the plane is the largest. This amounts to choosing maximum margin of separation. SVM gives best results for text classification because of sparse nature of text.

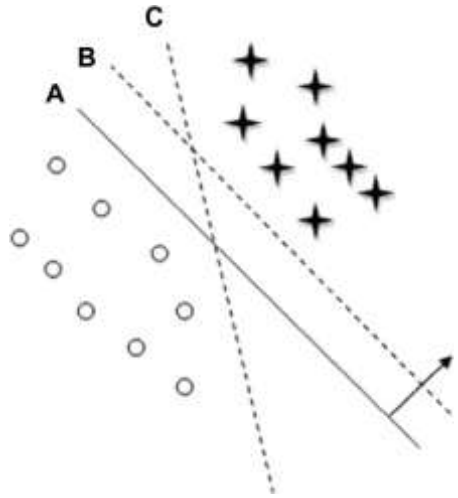


Figure 4.3.3 Support Vector Machine for Classification

Pseudocode for SVM

Require: **X** and **y** loaded with training labelled data

α ⇐ 0 or **α** ⇐ partially trained SVM // α_i and α_j are *Lagrange Multiplier*.

C ⇐ some value (10 for example) //Soft Margin Parameter

repeat

for all $\{x_i, y_i\}$, $\{x_j, y_j\}$ **do**

Optimize α_i and α_j

end

until no changes in α or other resource constraint criteria met

Ensure: Retain only the support vectors ($\alpha_i > 0$)

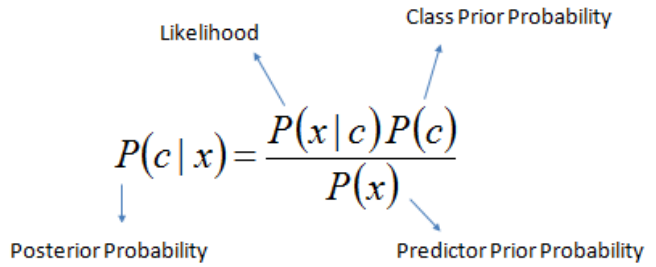
- **Naïve Bayes –** This classifier is the simplest and the most commonly used. It works well for text classification since it finds posterior probability of class, based on distribution of features in the document. Usually, features are extracted using Bag of Words (BOW). The classifier naively assumes that features are independent of each other.

Bayes Theorem is used to predict the probability of an observation belonging to a particular label given a set of features:

P(label/features) = [P(features/label) * P(label)]/P(features)

P(label) is the probability of observing a particular label. Given a label, P(features|label) is the likelihood that the feature belongs to that label. P(features) is the prior probability that a given feature has occurred. We naively assume that all features are independent of each other. The resultant equation can be written as follows:

P(label|features) =[P(label)*P(f1|label)*…P(fn|label)]/P(features)



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 4.3.4 Naïve Bayes working

Pseudocode for Naïve Bayes using discrete-valued feature-

Learning phase- Given a training set S,

For each target value of c_i ($c_i = c_1,\ldots c_L$)

$P(C = c_i)$ ← estimate $P(C = c_i)$ with examples in S

For every feature value x_{jk} of each feature X_j ($j=1,\ldots n$; $k=1,\ldots N_j$)

$P(X_j = x_{jk} \mid C = c_j)$ ← estimate $P(X_j = x_{jk} \mid C = c_i)$ with examples in S

Output- conditional probability tables; for X_j, NxL elements

Test phase- Given an unknown instance $X^* = (a_1, \ldots, a_n)$

Look up tables to assign the label c^* to X^* **if**

$[P(a_1|c^*)\ldots P(a_n|c^*)]P(c^*) > [P(a_1|c)\ldots P(a_n|c)]P(c)$ $(c \neq c^*, c=c_1,\ldots,c_L)$

- **K Nearest Neighbour (KNN) -** The k-nearest neighbours algorithm (k-NN) is used both for classification and regression problems. The input has k closest neighbours in the training space. The output depends on whether KNN is used for regression or classification:

I. In classification problem, the object is assigned to one of the available classes. The class to which the object is assigned is the class which is most common among its k neighbours. If k is 1, then the object is assigned to the class of its single nearest neighbour.

II. In regression problem, the output is a continuous property value of the object. The value to be assigned to the object is obtained by averaging the property values of its k neighbours.

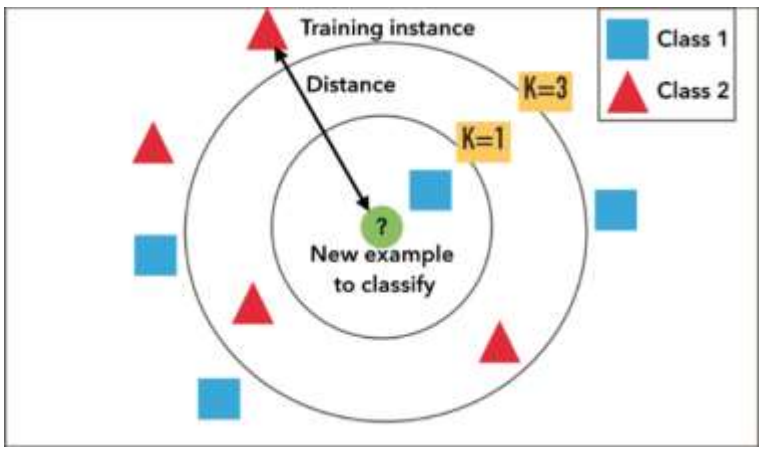


Figure 4.3.5 K Nearest Neighbour algorithm

Pseudocode for k-Nearest Neighbour-

Classify (**X**,**Y**,x) // **X**: training data, **Y**: class labels for X, x: unknown sample

for i=1 to m do

Compute distance $d(X_i,x)$

end for

Compute set *I* containing indices for the k smallest distances $d(X_i,x)$

return label that appears maximum number of times for {Y where i € *I* }

- **Gradient Boosting –** This is used mainly for regression and classification problems. It uses an ensemble of weak prediction models, particularly decision trees to produce a resultant prediction model. The intuition is to strengthen the weak prediction models and make them better repetitively by leveraging the patterns in residuals. Once residuals are devoid of any pattern that can be modelled, we stop modelling residuals for preventing overfitting.

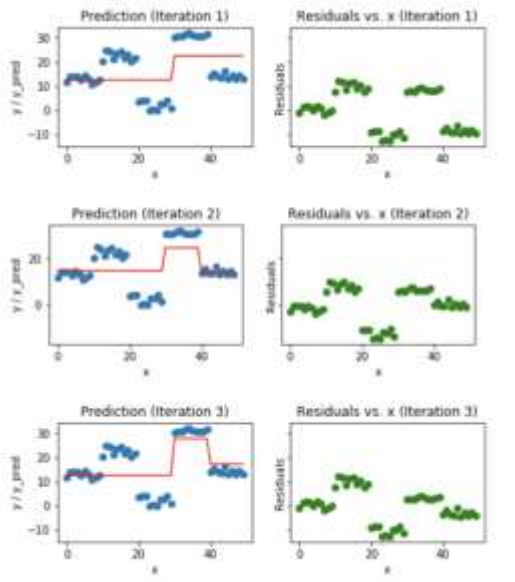


Figure 4.3.6 Gradient Boosting algorithm

Pseudocode for gradient boosting-

1) Let x be the input and y be the output. Fit a linear regressor or a decision tree using x and y
2) Find out error residuals i.e. the actual target value minus the predicted target value.

   e1= y - y_predicted1
3) Take a new model and fit it on error residuals. Let this be e1_predicted.
4) Add the predicted residuals obtained in the previous step to the earlier predictions

   y_predicted2 = y_predicted1 + e1_predicted
5) With the residuals that are still left, fit another model i.e. **[e2 = y - y_predicted2]** and keep repeating steps 2 to 5. We stop when the algorithm overfits or the sum of the residuals reach a constant value.

4.4. Results from Machine Learning

The above machine learning algorithms used for sentiment classification purpose are then compared on the basis of the accuracy they achieve on the testing data. The accuracy here means the fraction of testing samples that each algorithm was able to classify correctly into the two predefined classes i.e. positive and negative. The results obtained via the machine learning approach are summarised below:

	Unigrams (n=1)	**Bigrams (n=2)**	**Trigrams (n=3)**	**Quadrigrams (n=4)**
Support Vector Machine (SVM)	**81.4%**	**69.93%**	**67.78%**	**67.38%**
Naïve Bayes	**82.85%**	**57.73%**	**45.6%**	**47.74%**
KNN	**70.65%**	**67.7%**	**67.7%**	**67.38%**
Gradient Boosting	**78.35%**	**68.32%**	**68.32%**	**67.38%**

Table 4.4.1 Results obtained from Machine Learning

In order to obtain intactness of results, the data is shuffled randomly multiple times to get a new training and testing set each time. The partitioning of the training and testing data is also varied from 60:40 to 70:30 to 75:25. The results obtained after each shuffling and partitioning are averaged to get the final results which have been noted in the above table. The results show that unigrams and bigrams perform better with machine learning algorithms than trigrams and quadrigrams. Larger values of n in n-grams tend to lower the accuracy. The best results were obtained using unigram method of feature extraction and Naïve Bayes machine learning classifier. A graphical plot of the results has been included in the Appendix 2.

4.5. Using Context

The machine learning approach we used above doesn't take into account the context of the words or features it is using. This tends to limit the accuracy of this approach. Without context, a tweet like "He is really good at cheating" can be classified as positive because of the presence of the term 'good' in it. It is only when appropriate context is taken, the above tweet can be categorised as negative since the word 'cheating' is a negative polarity word. So, context of each word helps in categorising its polarity better and hence improving the accuracy of regular sentiment analysis that we

implemented in the above sections. We find the context of each word of the tweet by finding the words that co-occur with it and then judge the sentiment polarity of the word by taking into account the polarities of the co-occurring words. This approach is different as it doesn't assign fixed and static prior sentiment polarities to words. Instead, it takes into account the co-occurrence patterns of words in different contexts in tweets to capture their contextual semantics and update their pre-assigned strength and polarity. The main idea behind contextual semantics is- "You shall know a word by the company it keeps". This suggests that the words that co-occur in a given context tend to have certain relation to each other which if captured can greatly improve the accuracy of sentiment analysis. This is explained in the next section.

4.6 Lexicon

For our lexicon based approach, we tried two major lexicons: SentiWordNet and VADER. A brief description of these lexicons and their characteristics are given below:

4.6.1 WordNet and SentiWordNet

WordNet is a lexical resource that groups words of the English language into groups of synonyms called synsets. In addition to this, it also specifies relationships among various synsets and their definitions and examples. The semantic relationships connecting various synsets include hypernyms, hyponyms, meronyms, holonyms etc. Therefore, it is often used as a combination of dictionary and thesaurus.

SentiWordNet has been derived from WordNet.

SentiWordNet= WordNet + Sentiment Information

SentiWordNet assigns sentiment information to each WordNet sysnets. In other words, it assigns three scores to each synset. Each synset has a positivity, negativity and objectivity score that tells how positive, negative and objective the terms contained in the synset are. For each synset s,

pos(s)=positivity score of synset s

neg(s)=negativity score of synset s

obj(s)=objectivity score of synset s

pos(s)+neg(s)+obj(s)=1

The SentiWordNet lexicon was made by training a set of eight different ternary classifiers, each differing in training set and learning algorithm. The final scores to the synsets are assigned as follows:

P score = Total no. of classifiers stating positive/8

N score =Total no. of classifiers stating negative/8

O score =Total no. of classifiers stating objective/8

4.6.2 VADER Sentiment Analysis

VADER (**V**alence **A**ware **D**ictionary and s**E**ntiment **R**easoner) is a lexicon that has been specifically built for social media purposes. It is used to obtain polarity indices for a given word. VADER also performs well in handling emoji's, acronyms and slangs. It not only tells us about the polarity of the sentiment but also gives a sense of how strong the sentiment expressed is. It outputs four scores- positive (pos), negative (neg), neutral (neu) and compound score i.e. the overall score. The Positive, Negative and Neutral scores represent the proportion of text that falls in these categories. The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive). The compound score metric is described next:

- **Positive sentiment:** compound score>=0.05

- **Neutral sentiment:** (compound score > -0.05) and (compound score < 0.05)

- **Negative sentiment:** compound score <= -0.05

We use the above compound score metric in our implementation.

Example: We calculated sentiment for the word "NICE" using VADER and the results were as follows:

Sentiment Metric	Score
Positive	1.0
Negative	0.0
Neutral	0.0
Compound	0.4215

Table 4.6.2.1. Sentiment Metric in VADER Analysis for word “NICE”

Following the compound score metric, we can infer that the word “NICE” has an overall positive sentiment since the compound score is way greater than 0.05.

Example: We can also put the entire tweet sentence in VADER. For the tweet: “*This phone is super cool!!*”, the results obtained through VADER are:

Sentiment Metric	Score
Positive	0.674
Negative	0.326

Neutral	0.0
Compound	0.735

Table 4.6.2.2. Sentiment Metric in VADER Analysis for the tweet: "*This phone is super cool!!*"

The results indicate that the tweet is 67% positive, 32% negative and 0% neutral. The compound score for the above tweet is 0.735, hence the tweet can be categorised as positive.

It is safe to say that VADER lexicon is able to analyse social media data effectively for the purpose of sentiment analysis. VADER analysis is based on certain key points as follows:

1. **Punctuation:** The use of an exclamation mark **(!)**, increases the strength of the semantic orientation. For example, "The food here is good!" is more intense than "The food here is good." and an increase in the number of **(!)** increases the strength accordingly.

2. **Capitalization:** Using upper case letters to express a sentiment related word increases its intensity. For example, "The food here is GREAT!" conveys more intensity than "The food here is great!"

3. **Degree modifiers:** They are also called intensifiers. They impact the sentiment intensity by either increasing or decreasing it. For example, "The service here

is extremely good" is more intense than "The service here is good", whereas "The service here is marginally good" reduces the intensity.

4. **Conjunctions**: Use of conjunctions like "but" signals a shift in sentiment polarity, with the sentiment coming after the conjunction being more powerful. "The food here is great, but the service is horrible" has mixed sentiment, with the latter part dominating the overall sentiment polarity.

5. **Preceding Tri-gram:** By studying a tri-gram preceding a sentiment feature, we can get to know that more than often negation flips the polarity of the text. A negated sentence would be "The food here isn't really all that great".

Besides having the above mentioned characteristics, VADER has numerous advantages when compared to traditional methods of sentiment analysis:

- It works well on social media text and generalises to multiple domains.

- VADER doesn't require training data. It is constructed from a general, valence-based sentiment lexicon.

- It is suitable for streaming data

- It does not have any major speed-performance trade off.

4.6.3. Comparision between SentiWordNet and VADER Sentiment Analysis

To choose a suitable lexicon for our project, we carried out a comparative study between the two lexicons explained above, namely SentiWordNet and VADER lexicon. The comparison has been summarised below:

Basis	**SentiWordNet**	**VADER Sentiment Analysis**
Description	Construction of a lexical resource for sentiment analysis based on WordNet. Synonym set or synset comprises of adjectives, nouns, adverbs etc. grouped together and associated with 3 polarity score for each word	Human validated sentiment analysis method used for twitter and social media contexts. VADER was created from a generalizable, valence band, human curated gold sentiment lexicon
Outputs	Provides positive, negative and objective scores for each word in the range of 0.0 to 1.0. Polarity determined by aggregating the three constituent scores using different aggregation techniques.	Provides positive, negative, neutral and compound score for each word. Compound score is used to determine the final polarity.
Validation	Validates the proposed dictionary with comparisons with other dictionaries but also uses human validation of the proposed lexicon	Validation is done using datasets like Twitter, Movie Reviews, Technical Product reviews, NYT, User Opinions
Techniques employed	It uses Lexicon and Machine	It uses a lexicon having a

	learning with lexicon size of 117,658	size of 7,517

Table 4.6.3.1. Comparision between SentiWordnet and VADER Sentiment Analysis

We wanted to obtain a single metric value to indicate the swing of a word between extremely positive and extremely negative sentiment range. In other words, it was imperative for us to get that single numerical estimate that gives the overall polarity of a word. Since SentiWordNet outputs three scores, it was difficult for us to aggregate them for the purpose of calculating the overall sentiment score. To overcome this problem, we used VADER Sentiment Analysis for score assignment, since VADER gives a single value with different cut off points indicating positive, negative, neutral, extremely positive and extremely negative entities.

4.7 Co-occurrence of words

In this section, we explain how we used co-occurrence patterns of words to find contextual semantics for sentiment analysis. In this method, sentiment polarities assigned to the words aren't fixed or static. Rather they change based on context and semantics. This approach allows for entity-level sentiment detection which analyses the sentiment polarities towards specific entities, say Taylor Swift or Starbucks. It also allows for tweet-level sentiment classification that determines the sentiment orientation of the overall tweet.

The main notion behind this approach is that the sentiment orientation of the word is not static or fixed but rather constantly changing according to its context. For example, most of the present-day implementations of sentiment analysis fail to classify the following tweet, "*#Syria: ISIS execution continues with a smile*" since the word *smile* has a positive sentiment orientation even though here it has been used in a negative sense, with the word *execution.* So, in order to find the sentiment of the target word like *smile,* we need to find all the co-occurring words i.e. words that occur with the word "*smile*" in the dataset and then determine its sentiment orientation. The words that co-occur in a given

context tend to have certain relation to each other which if captured can greatly improve the accuracy of sentiment analysis. The process of finding co-occurrences between words is summarised below. The target word is the word whose co-occurrence pattern has to be found out.

For very target term, we find all the co-occurring words for it in the entire dataset. We assign a specific position to each co-occurring word to get a pattern representation of the target word. Each position of the co-occurring word determines its sentiment influence towards the target word. These positions can be represented as an angle and a radius. The angle determines the prior sentiment of the co-occurring word as given by the lexicon VADER and the radius represents the strength of correlation between the target and co-occurring words. The angle Θ is calculated as:

$$\Theta = \text{Prior Sentiment from lexicon VADER} * \pi$$

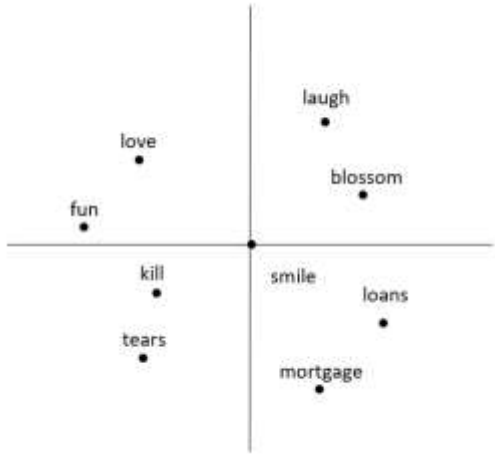


Figure 4.7.1: Co-occurrence pattern of a target word "*SMILE*" using context information

The prior sentiment value will range anywhere from -1 to 1, hence Θ will range from -π to π. The region from 0 to π captures the positive sentiment (0 being neutral,

π being extremely positive). Similarly, the region from 0 to -π captures the negative sentiment region (0 being neutral, -π being most negative). Terms in the upper two quadrants have positive sentiments with the upper left quadrant having a stronger positive sentiment polarity than the upper right one. Similarly, the bottom two quadrants have negative sentiment polarities, with the bottom left being more negative. The radii range from 0 to 1 which indicate how important the context or co-occurring terms are for determining the polarity of the target word. A large radius implies that the corresponding context term has more influence in determining the overall polarity of the target term.

After finding the co-occurrence pattern of the target word, we find the geometric median of all the points obtained from the co-occurring words to get the overall polarity of the target word. This has been described next.

4.8 Geometric Median

We find the geometric median of the target word to find its overall polarity. The geometric median is a point capturing the overall sentiment and strength of the target word. The geometric median of a set of points is defined as the point to which the sum of Euclidean distances of the points is the minimum. The position or the quadrant in which the final geometric median lies gives the overall polarity of a word. The quadrants have the same polarities as defined in the above section.

After calculating the geometric median for each word in a tweet, we calculate the overall geometric median of the tweet i.e. the geometric median of all geometric medians. This gives the net polarity w.r.t the overall tweet i.e. pertaining to the entity being discussed in the tweet.

The process has been explained for the following tweet:

"*Bad weather causes problem!!*"

For the words, *bad, weather, causes* and *problem,* we first find their co-occurring patterns. This is followed by finding their respective geometric medians, let them be denoted by GM_{Bad}, $GM_{weather}$, GM_{causes} and $GM_{problem}$. Using the above geometric medians of each word in the tweet, we calculate the overall geometric median of the tweet. Depending on the position of the overall geometric median, we determine the sentiment of the tweet.

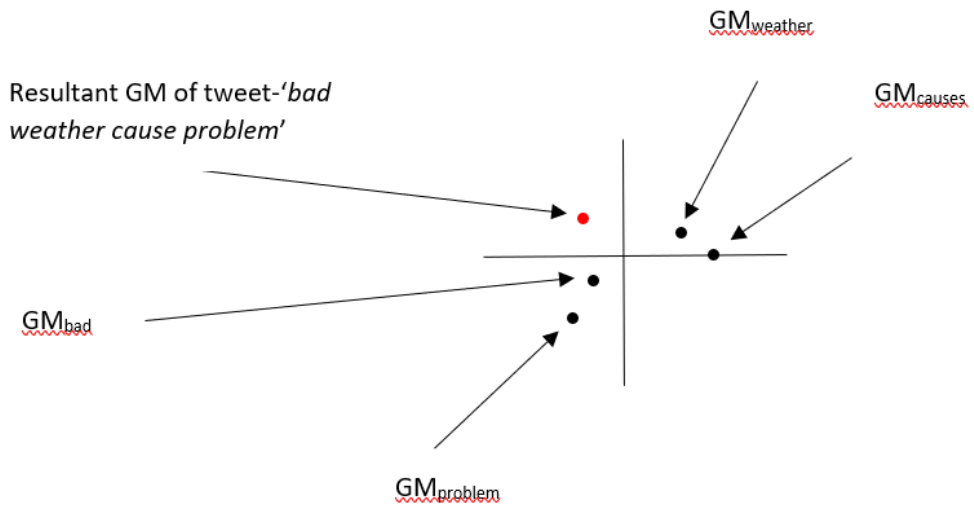


Figure 4.7.2: Finding resultant geometric median (GM) of a tweet

We use the Weiszfeld algorithm to find the geometric median of a collection of points. The algorithm described as follows:

The Weiszfeld algorithm is used to find the geometric mean of a collection of points in Euclidean space. It is usually used with L1 norm, but has various generalised versions of it as well.

The input given to the algorithm is a set of n dimensional data points. The algorithms starts by assigning a candidate median as the mean of all data points. The candidate median is a starting point for the algorithm to run from. The number of iterations are fixed apriori. Before each iteration, the denominator is fixed as the sum of Euclidean distances of all data points from the candidate median. For each iteration, a fixed relation is calculated for all the data points and the candidate median is updated. The algorithm stops when the change in the candidate median is less than a specified value €. After fixed number of iterations, we get the final geometric median point. Pseudocode of Weiszfeld algorithm is defined as follows:

Pseudocode for Weiszfeld Algorithm:

Fix num_iterations ← some constant c

For a set of t n-dimensional data points [p_1,p_2,……..p_t]

candidate_median ← mean of all t points

denominator ← 0

For i^{th} iteration in num_iterations:

    For p_i in data points:

        denominator ← denominator + 1/Euclidean_distance(p_i, candidate_median)

    next_median ← 0

    For p_i in data points:

        next_median ← next_median + (p_i * 1/ Euclidean_distance (p_i, candidate_median))/ denominator

    candidate_median ← next_median

4.9. Putting it together

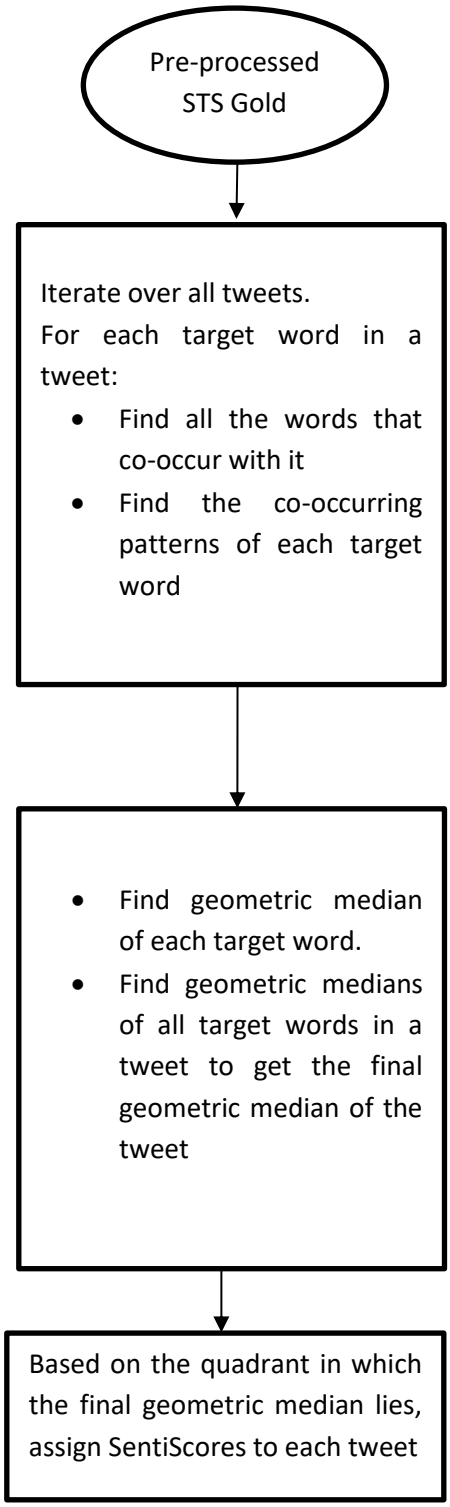


Figure 4.9.1. The process of context based sentiment analysis

1) After pre-processing the entire STS-Gold dataset, we go over the tweets in our dataset one by one.

2) For each word (*target_word*) in the tweet, we perform the following procedure:

   2.1) Scan the entire dataset to find all the words that co-occur together with the *target_word*. Let these be the *co-occurring_words.* These are those words that occur together with the *target_word* in a tweet and are not stop words at the same time.

   2.2) Create a dictionary that stores the *co-occurring_words* found in the above step and their frequency i.e. the number of times each *co-occurring_word* occurs together with the *target_word* in the same tweet. Every element of this dictionary is a key-value pair. The dictionary looks something like this:

   {

   (*co-occurring_word_1* --> f1)

   (*co-occurring_word_2* --> f2)

   …..

   (*co-occurring_word_*N_c --> fn)

   }

   Where f1, f2……fn are the frequencies corresponding to the N_c *co-occuring_words*.

Each *target_word* has a dictionary like the one shown above.

2.3) After creating the dictionary, for each *target_word*, its subsequent co-occurrence pattern is found out. The radius is calculated to determine its position. The radii r for each *co-occurring_word* is calculated as follows:

$$\text{Ri} = \text{R}\left(co - occurring_{word_i}, target_{word}\right) = \text{f}\left(co - occurring_{word_i}, target_{word}\right) * \log\frac{N}{N_c} \quad (4.1)$$

where

f(*co-occurring_word_i*,*target_word*) = frequency of occurrence of *co-occurring_word_i* with *target_word*

N= total number of words in the dataset

Nc = Total number of *co-occurring_words*

The radii is also called as the degree of correlation (TDOC) between *co-occurring_words* and *target_word*. Each R_i is then arranged in a vector to obtain **R** vector.

2.4) For finding the angle theta, Θ, we find the prior polarity of each of the N_c *co-occurring_words* using the VADER lexicon and then multiply it with π.

$$\theta_i = \text{Prior Sentiment of } co - occurring\_word\_i \text{ from VADER lexicon } * \pi \quad (4.2)$$

Each Θ_i is then arranged in a vector to obtain **Θ** vector. The co-occurrence pattern representation of the *target_word* is hence made (Figure 4.7.1).

2.5) The position of the *co-occurring_words* with their Cartesian coordinate *(x,y)* is calculated as

$$x = R\left(target_{word}, co - occurring_{word_i}\right) * \cos\theta_i \quad (4.3)$$

$$y = R\left(target_{word}, co - occurring_{word_i}\right) * \sin\theta_i \quad (4.4)$$

where

R(target_word, co-occurring_word_i)= the degree of correlation (TDOC)

Θ_i = angle of the co-occurring term

x represents the sentiment strength

y represents the sentiment polarity.

3) After determining the position of the *co-occuring_words* of the *target_word*, we want to find the final sentiment polarity of it. This is done by finding the geometric median of the *co-occuring_words*. The geometric median of a collection of points, as explained above is the point from which the sum of Euclidean distances to all points is minimum. This point is found using the Weiszfeld algorithm.

For a set of n points (p_1,p_2,…p_n) , geometric median k with coordinates (x_t,y_t) is defined as

$$k = \arg min_{k\in\mathbb{R}^2} \sum_{i=1}^{i=n} ||p_i - k||_2 \quad (4.5)$$

The point k captures the sentiment polarity through its y coordinate and the sentiment strength through the x coordinate of all the target term.

4) The geometric median of all the words in a tweets are found out using the above steps. Once all the geometric medians are found out, the final geometric median is obtained by finding out their geometric median. Let this be Final_GeometricMedian.

5) Depending on the quadrant in which the Final_GeometricMedian lies, the overall sentiment polarity of the tweet is decided. This polarity is call **SentiScore.**

The **SentiScores** range from 0 to 4, 0 being the most negative and 4 being the most positive.

SentiScore	**Sentiment**
0	Extremely Negative
1	Negative
2	Neutral
3	Positive
4	Extremely Positive

Table 4.9.1. Polarity Score and Sentiment Assignment

The assignment of **SentiScores** follow the below rules:
I quadrant- tweet is positive- Score is 3
II quadrant- tweet is positive- Score is 4
III quadrant- tweet is positive- Score is 0
IV quadrant- tweet is positive- Score is 1

If the geometric median lies on the x-axis, the tweet is neutral and hence, a **score** of 2 is assigned to it.

4.10. Aggregation

Aggregation is the stage where we combine the results obtained from machine learning and context approaches to get the final sentiment scores of the tweets. There are several ways of aggregating two numerical values, in our case **MLScore** and **SentiScore**. Some of them have been summarised in the next section. The aim of this stage of the implementation is to get a indicative final score of the tweets that predict their true nature of sentiment polarities.

4.10.1. Different Aggregation Techniques

Aggregation Methods are the types of calculations which are used to group attribute values into a single metric for each dimension. There are different statistical aggregation methods to group two data values. Some of them are explained as follows-

1. **Sum** – Calculates the total value of a metric by adding the constituent values. This can be used for numbers and durations. This method cannot be used for multi-value attributes.

2. **Average/Mean** – Calculates the average value of the metric. This aggregation method can be used for numbers, dates, times and durations. This method cannot be used for multi-value attributes.

3. **Median** – It calculates the median value for the metric. This aggregation method can be used for numbers, dates, times and durations, but not for multi-value attributes.

4. **Weighted Mean** – In this method, various weights are assigned to different metrics and their mean value is calculated.

$$\text{weighted mean} = \frac{\sum \text{wx}}{\sum \text{w}} \tag{4.6}$$

Where the weight is denoted by w and the data metric by x

5. **Minimum**- Selects the minimum value for the metric. This aggregation method can be used for numbers, dates, times and durations but not for multi-value attributes.

6. **Maximum** - Selects the maximum value for the metric. This aggregation method can be used for numbers, dates, times and durations but not for multi-value attributes.

7. **Standard Deviation-** Calculates the standard deviation for the metric values. This aggregation method can be used for numbers, dates, times and durations but not for multi-value attributes.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (4.7)$$

Where μ= mean

N= total number of values

x_i = data value

8. **Variance**- Calculates the variance (square of the standard deviation) for the metric values. This aggregation method can be used for numbers, dates, times and durations but not for multi-value attributes.

The various methods discussed above can be used for aggregation of numerical quantities.

4.10.2. Proposed Aggregation Methodology

The above techniques, though effective, were not particularly suitable for our implementation.

In our implementation, we need to aggregate two kinds of data: Machine learning labels, mainly 0(negative) and 4(positive), and context based labels, ranging from 0-4(from extremely negative to extremely positive), as discussed in section 4.9. These two kinds of data have different nature; while the range for both of them is same, their step size is different. This makes them incompatible for aggregation methods like summation, minimum and maximum.

For methods like mean or weighted mean, when applied, the misclassification error was extremely high, which led to discarding of the above approaches. Also, methods like standard deviation and variance are used to measure the spread of data values or variation in the values from the mean value, and hence are unsuitable for our set of values.

Hence due to the limitations of the above aggregation techniques, we have defined our own aggregation method for the final step of our project.

Our technique involves conversion of **MLScores** and **SentiScores** into angular values, followed by summing them up. This has been explained next.

As discussed earlier, in our proposed methodology, we will combine the polarities of both machine learning approach and context based approach to arrive at a concluded polarity. This enables us to take into account the advantages of both the approaches. Machine learning approach helps us to deal with domain independent data while context based approach allows us to take context into account for a better accuracy of sentiment classification and increase the sentiment labels from two to five.

1) The first step is to convert **SentiScore** into a planar angle. This is accomplished by taking $\tan^{-1}$ of the ratio of the y-coordinate to the x-coordinate of the geometric median of the tweet obtained in the previous stages of the implementation. Formally, the angle Θ_{senti} or SentiTheta is expressed as,

$$\Theta_{senti} = \tan^{-1}\left(\frac{y\ coordinate\ of\ the\ SentiMedian\ of\ tweet}{x\ coordinate\ \ of\ the\ SentiMedian\ of\ tweet}\right) \quad (4.8)$$

2) Next, we convert the **MLScore** into angle. If the predicted machine learning polarity p is negative or 0, we map the MLScore onto $-\pi/4$ planar angle and if the polarity is positive or 4, we map it onto $+\pi/4$ angle. We can express the above statement as:

For predicted polarity p, angle Θ_{ml} or MLTheta will be,

$$\Theta_{ml} = \begin{cases} -\frac{\pi}{4} & \text{for } p = 0 \\ \frac{\pi}{4} & \text{for } p = 4 \end{cases} \quad (4.9)$$

Where p is predicted polarity obtained from machine learning.

3) Combining the two above angles, Θ_{ml} and Θ_{senti}, we output the total angle, θ_{total}, which based on its values, is divided into five sentiments.

$$\Theta_{total} = \Theta_{ml} + \Theta_{senti} \quad (4.10)$$

The final sentiment scores are assigned based on table shown below.

Final Sentiment Label	**Angle mapping**
Neutral	$-5° < \Theta_{total} < 5°$
Positive	$5° < \Theta_{total} < 90°$
Negative	$-90° < \Theta_{total} < -5°$
Extremely Positive	$90° < \Theta_{total} < 180°$
Extremely Negative	$-90° < \Theta_{total} < -180°$

Table 4.10.2.1. Final sentiment assignment based on angles

The neutral region is defined in terms of angles from -5° to +5°. The positive region is from +5° to +180° while the negative region is from -5° to -180°. The intensity of the positive and negative sentiment increases as the magnitude of the total angle increases.

After assigning the final sentiment polarities to the tweets, we find the **misclassification error** i.e. we match the actual polarity of the tweets in the STS Gold dataset as assigned by the creators of the dataset and the polarities that our implementation assigned to the tweets. If there is a mismatch between the two for a particular tweet, then we conclude that the tweet has been misclassified by our implementation.

CHAPTER 5

RESULTS

We performed the entire implementation twice with different subsets of test data each time. We divided the STS data in a 60:40 ratio twice after shuffling the entire dataset to make sure that we got two different test sets. The we extracted unigram features from the training data and trained our Naïve Bayes algorithm on it. After that, we obtained the predicted machine learning scores, **MLScores** for the test sets. Then we implemented our context based approach to find the sentiment score of each tweet using entity level sentiment analysis. This was followed by aggregating the **MLScores** and **SentiScores** of each tweet in the test sets.

The results obtained from the two rounds of implementation were as follows:

Ist round of implementation:

Error obtained by using only machine learning: 21.25% (Accuracy=78.75%)

Error obtained by using context along with machine learning: 16.4% (Accuracy=83.6%)

Improvement in accuracy: 4.9%

II[nd] round of implementation:

Error obtained by using only machine learning: 20.8% (Accuracy=79.2%)

Error obtained by using context along with machine learning: 15.2% (Accuracy=84.8%)

Improvement in accuracy: 5.7%

So, we can conclude that the improvement in accuracy is approximately 5-6% i.e. the misclassification error of tweets comes down by this range.

CHAPTER 6

CONCLUSIONS

We can conclude that taking context into consideration improves the accuracy of sentiment analysis. This is mainly because words change their meanings from one setting to another i.e. their meaning depends on the context in which they are used. Identifying this context is a precondition for effective sentiment classification. Machine learning alone cannot produce the required amount of accuracy since this approach depends on learning only features and then using these features for making predictions on testing data.

Machine learning algorithms try to guess a static relationship between the input i.e. the features (unigrams in our case) and the final labels (sentiment polarities) that make up the training data and using this derived relationship make further predictions on the test set. This kind of an approach is often plagued with inaccuracies for language processing since languages do not have a fixed set of rules. They are dynamic and a host of interdependent factors like usage, context, constructs, semantics, discourse etc play a role in determining the underlying meaning of text. Therefore, machine learning alone doesn't suffice for sentiment classification task. Using context along with machine learning is the first step in understanding the nuanced interpretations of language text (tweets in this case). By using context, a tweet that was classified as just negative by our machine learning approach can swing into the neutral or positive class if it has a strong sentiment orientation towards these regions. Similarly, a tweet classified as just positive by machine learning algorithm can swing into the neutral or negative region, hence giving us more accuracy.

Apart from enhancing the accuracy, using context also helps us in achieving a fine-grained sentiment analysis. We could increase the number of classes from two (positive and negative classes as given in the dataset) to five (extremely positive, positive, neutral, negative and extremely negative classes) to assess the strength of the sentiment expressed in the tweets and hence identify how strong the underlying emotions are.

The above passages mention some ways by which we have tried to perform a better sentiment analysis. There is immense potential behind context based sentiment analysis. A lot of creative techniques can be employed in improving the accuracy of sentiment classification of twitter data further. This involves further research and intensive study of the twitter domain. Context based sentiment analysis is a developing research area with the capacity to develop into something completely astounding considering the amount of potential this field has. Our dissertation is a small contribution to this exciting journey.

6.1. Limitations

Our implementation faces a few challenges at the time of writing this dissertation. These are research areas in themselves and have been described below:

- **Negation handling-** Text doesn't always have strong opiniated words to express sentiments, specifically negative sentiments. For example, the tweet "*This iPhone has a really short battery life*" conveys a negative aspect about the iPhone without using any opiniated words. These kinds of tweets are very difficult to detect and are often missed. Other than this, sometimes there are not enough linguistic features available for the algorithm to classify the tweets as negative. Negation handling is a challenge that our approach faces.

- **Sarcasm detection-** This is another challenge that our implementation faces. Tweets like '*This is the best day of my life! Lost my job and my dog died*' or '*Oh!*

How wonderful! My phone stopped working again' will be taken as positive tweets. To detect sarcasm, we have to able to understand how facts relate to events being talked about. The contradiction between objective polarity (which is negative) and sarcastic remarks conveyed by the author (which are positive) has to be comprehended for detection of sarcasm.

6.2. Future Scope

There is scope for incorporating a lot of other concepts and techniques in our present implementation. The future scope of our project has been described below:

- Machine learning approaches usually needs a lot of labelled data in their training phase. An alternative to this is to use a combination of deep learning and sentiment analysis techniques. Deep learning algorithms like convolutional neural networks, recurrent neural networks and deep neural networks have the ability to learn new features automatically and hence have much less data requirements.
- Instead of using co-occurrence patterns for finding context of a word, we can use other approaches like ontologies, shifter clues, collocational features etc and compare the accuracy obtained using these methods with the already implemented co-occurrence method.
- An even finer grained sentiment classification can be carried out by increasing the number of classes. These classes can be moderately positive, positive, very positive, extremely positive for positive tweets. The same can be done for negative and neutral classes as well. This would provide an even greater insight into user attitudes and emotions. To achieve this, we need a higher degree of variance in the co-occurrence patterns and hence sentiment polarity scores of tweets in the dataset.

- For getting the polarity of words, instead of using inbuilt lexical resources like VADER or SentiWordNet, we can also make our own lexicon using dictionary-based and corpus-based approaches. This can provide greater coverage of words and will be more suitable for domain specific applications.

APPENDIX

d	tweet	polarity
1.468E+09	the angel is going to miss the athlete this weekend	0
2.323E+09	It looks as though Shaq is getting traded to Cleveland to play w/ LeBron.	0
1.468E+09	@clarianne APRIL 9TH ISN'T COMING SOON ENOUGH	0
1.99E+09	drinking a McDonalds coffee and not understanding why someone wou	0
1.989E+09	So dissapointed Taylor Swift doesnt have a Twitter	0
1.468E+09	Wishes I was on the Spring Fling Tour with Dawn & neecee Sigh G'	0
1.965E+09	got a sniffle, got the kids and hubby just left to work in Sydney for the w	0
1.881E+09	i've only been in sydney for 3 hrs but I miss my friends especially @ktja	0
1.754E+09	xboxtweet not working again	0
1.98E+09	R.I.P to lebron/kobe puppet commercials...	0
1.755E+09	Allergies sucks sometimes. Theres a super adorable 9 month old beagle	0
1.966E+09	has a broken iphone	0
2.206E+09	Line at McDonalds was too long so I can't get my sausage biscuit on	0
1.688E+09	@stephnewby there is a virus going around congestion, throw up, &am	0
1.964E+09	I scratched my iPod	0

Figure A1. STS-Gold dataset

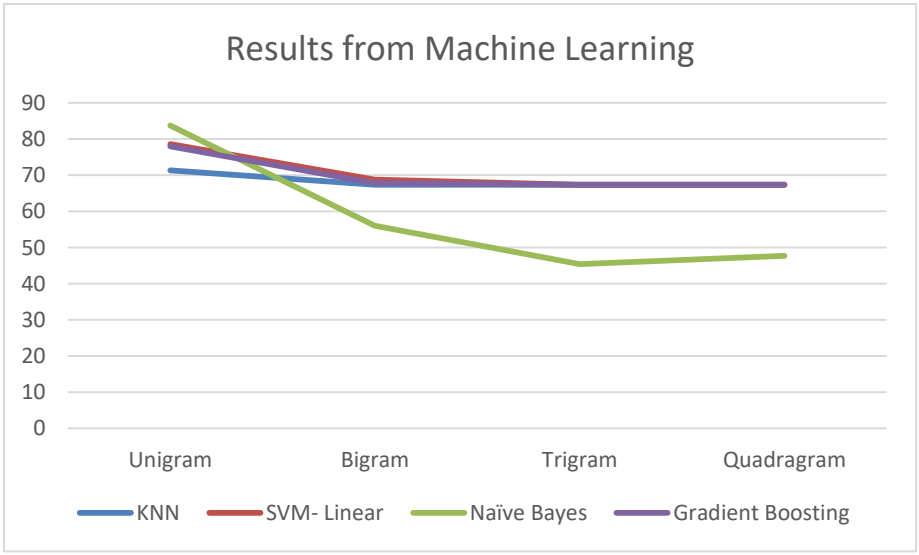


Figure A2. Results from Machine Learning

Figure A3. Assigning Sentiment Polarities using VADER

REFERENCES

[1] Anuja P Jain ; Padma Dandannavar, “Application of machine learning techniques to sentiment analysis”, 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)

[2]. Deepak Singh Tomar, Pankaj Sharma, “A Text Polarity Analysis Using Sentiwordnet Based an Algorithm”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1), 2016, 190-193.

[3]. Walaa Medhat, Ahmed Hassan, Hoda Korashy, “Sentiment analysis algorithms and applications: A survey”, Ain Shams Engineering Journal, Volume 5, Issue 4, December 2014, Pages 1093-1113.

[4]. Gilad Katz, Bracha Shapira, Nir Ofek, Yedidya Bar-Zev, “ A Context Based Approach for Text Classification”, Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2015: Advances in Knowledge Discovery and Data Mining pp 27-38.

[5]. Bo Pang, Lillian Lee, Shivakumar Vaithayanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques”, Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol.10, 2002,pp. 79-86.

[6]. Bing Liu, Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012.

[7]. M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis”, Computational Linguistics, vol. 37, 2011, pp. 267-307.

[8]. Ji Fang and Bi Chen, “Incorporating Lexicon Knowledge into SVM Learning to Improve Sentiment Classification”, In Proceedings of the workshop on Sentiment Analysis where AI meets Psychology(SAAIP), pages 94-100,2011.

[9]. Neethu M S, Rajashree R, “Sentiment Analysis in Twitter using Machine Learning Techniques”, 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT).

[10]. Brendan O’Connor, Ramnath Balasubramanyan,Bryan R. Routledge, Noah A. Smith, "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series ", Icwsm 11 (122-129), 1-2.

[11]. Johan Bollen, Alberto Pepe, Huina Mao, "Modelling public mood and emotion: Twitter sentiment and socio-economic phenomena", Published 2011 in ICWSM.

[12]. A. Tumasjan, T. O Sprenger, P. G Sandner, I. M Welpe (2010), "Predicting elections with Twitter: What 140 characters reveal about political sentiment", Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media.

[13]. L Chen, W Wang, M Nagarajan, S Wang, AP Sheth, "Extracting Diverse Sentiment Expressions with Target-Dependent Polarity from Twitter", ICWSM 2 (3), 50-57.

[14]. E Junqué de Fortuny, T De Smedt, D Martens, "Media coverage in times of political crisis: A text mining approach", An international Journal Archive. Vol 39, Issue 14, October 2012, Pages 11616-11622.

PLAGIARISM REPORT

major_sem8_final.docx

ORIGINALITY REPORT

14%	10%	12%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	stp.lingfil.uu.se Internet Source	2%
2	ijarcce.com Internet Source	1%
3	SpringerBriefs in Cognitive Computation, 2016. Publication	1%
4	labs.imaginea.com Internet Source	1%
5	ece.usu.edu Internet Source	1%
6	pdfs.semanticscholar.org Internet Source	1%
7	ijsrset.com Internet Source	1%
8	www.stat.ncsu.edu Internet Source	1%
9	Hunaida Awwad, Adil Alpkocak. "Performance Comparison of Different Lexicons for	1%

Sentiment Analysis in Arabic", 2016 Third European Network Intelligence Conference (ENIC), 2016
Publication

10	Katarzyna Tarnowska, Zbigniew W. Ras, Lynn Daniel. "Recommender System for Improving Customer Loyalty", Springer Science and Business Media LLC, 2020 Publication	<1%
11	hyse.org Internet Source	<1%
12	icwsm.org Internet Source	<1%
13	K.Sai Vishnu, T. Apoorva, Deepa Gupta. "Learning domain-specific and domain-independent opinion oriented lexicons using multiple domain knowledge", 2014 Seventh International Conference on Contemporary Computing (IC3), 2014 Publication	<1%
14	dspace.bracu.ac.bd:8080 Internet Source	<1%
15	"A Practical Guide to Sentiment Analysis", Springer Nature, 2017 Publication	<1%
16	ermt.net	

	Internet Source	<1%
17	Mahima Goyal, Vishal Bhatnagar. "chapter 10 A Classification Framework on Opinion Mining for Effective Recommendation Systems", IGI Global, 2017 Publication	<1%
18	"PRICAI 2014: Trends in Artificial Intelligence", Springer Science and Business Media LLC, 2014 Publication	<1%
19	www.slideshare.net Internet Source	<1%
20	boa.unimib.it Internet Source	<1%
21	repository.liv.ac.uk Internet Source	<1%
22	Sunday Adewale Olaleye, Ismaila Temitayo Sanusi, Jari Salo. "Sentiment analysis of social commerce: a harbinger of online reputation management", International Journal of Electronic Business, 2018 Publication	<1%
23	ijarcsse.com Internet Source	<1%

24	JEONG-MI CHO, JUNGYUN SEO, GIL CHANG KIM. "Verb sense disambiguation based on dual distributional similarity", Natural Language Engineering, 1999 Publication	<1%
25	Amogh Madan, Ridhima Arora, Nihar Ranjan Roy. "Chapter 47 Sentiment Analysis of Indians on GST", Springer Nature America, Inc, 2018 Publication	<1%
26	eprints.covenantuniversity.edu.ng Internet Source	<1%
27	Lei Zhang, Shuai Wang, Bing Liu. "Deep learning for sentiment analysis: A survey", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018 Publication	<1%
28	"Encyclopedia of Machine Learning and Data Mining", Springer Nature, 2017 Publication	<1%
29	Chinsha T C, Shibily Joseph. "A syntactic approach for aspect based opinion mining", Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), 2015 Publication	<1%
30	"Smart Trends in Information Technology and	

Computer Communications", Springer Science and Business Media LLC, 2016

Publication

<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		