Fall 2021

Mining Twitter Sequences of Product Opinions with Multi-Word Aspect Terms

Vinay Kiran Manjunath
University of Windsor

Mining Twitter Sequences of Product Opinions with Multi-Word Aspect Terms

By

Vinay Kiran Manjunath

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2021

Mining Twitter Sequences of Product Opinions with Multi-Word Aspect Terms

By

Vinay Kiran Manjunath

APPROVED BY:

D. Borisov

Department of Mathematics & Statistics

A. Biniaz

School of Computer Science

C. Ezeife, Advisor

School of Computer Science

September 1, 2021

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Social media platforms have opened doors to users' opinions and perceptions. The text remains the most popular means of contact on social media, despite different means of communication (audio/video and images). Twitter is one such microblogging platform that allows people to express their thoughts within 280 characters per message. The freedom of expression has made it difficult to understand the polarity (Positive, Negative, or Neutral) of the tweets/posts. Given a corpus of microblog texts (e.g., "the new iPhone battery life is good, but camera quality is bad"), mining aspects (e.g., battery life, camera quality) and opinions (e.g., good, bad) of these products are challenging due to the vast data being generated. Aspect-Based Opinion Mining (ABOM) is thus a combination of aspect extraction and opinion mining that allows an enterprise to analyze the data in detail, saving time and money automatically.

Existing systems such as Hate Crime Twitter Sentiment (HCTS) and Microblog Aspect Miner (MAM) have been recently proposed to perform ABOM on Twitter. These systems generally go through the four-step approach of obtaining microblog posts, identifying frequent nouns (candidate aspects), pruning the candidate aspects, and getting opinion polarity. However, they differ in how well they prune their candidate features. HCTS uses Apriori based Association rule mining to find the important aspects (single and multi word) of a given product. However, the Apriori based system generate many candidate sequences which generates redundant candidate aspects and HCTS also fails to summarize the category of the aspects (Camera? Battery?). MAM follows the similar approach to that of HCTS for finding the relevant aspects but it further clusters the frequent nouns (aspects) to obtain the relevant aspects. However, it does not identify the multi-word aspects and the aspect category of a product.

This thesis proposes a system called Microblog Aspect Sequence Miner (MASM) as an extension of Microblog Aspect Miner (MAM) by replacing the Apriori algorithm with the modified frequent sequential pattern mining algorithm. The system uses the power of sequential pattern mining for aspect extraction in ABOM. The sentiments of the tweets are unknown, so we build our approach in an unsupervised learning manner. The input posts are first classified to identify those tweets which contain the opinion (subjective) to those that do not have any opinion (objective). Then we extract the Parts of Speech tags for the explicit aspects to identify the frequent nouns. The novel frequent pattern mining framework (CM-SPAM) is applied to segment the single and multi-word aspects which generates less sequences as compared to previous approaches. This prior knowledge helps us to operate a topic modeling framework (Latent Dirichlet Allocation) to determine the summary of most common aspects (Aspect Category) and their sentiments for a product. The findings demonstrate that the MASM model has a promising performance in finding relevant aspects with reduction of average vector size (cost of candidate/aspect generation) against the MAM and HCTS using the Sanders Twitter corpus dataset. Experimental results with evaluation metrics of execution time, precision, recall, and F-measure indicate that our approach has higher recall and precision than the existing systems.

Keywords: Aspect based opinion mining; Sequential pattern mining; Multi-word extraction; Topic modeling; Twitter opinion mining; Subjectivity classification

DEDICATION

I would like to dedicate this thesis to my parents, supervisor, internal and external readers, and my friends who have helped and supported to complete my graduate study at the University of Windsor.

ACKNOWLEDGEMENTS

My sincere appreciation goes to my parents Mr. M. S. Manjunatha and Mrs. Jayashree H R. Your love, faith and words of encouragement gave me the extra energy to see this work through.

I would like to express my sincere gratitude to my advisor Prof. Dr. Christie Ezeife for her unwavering support throughout my graduate study and constantly pushing to strive better. Thank you for your patience and effort spent reading all my thesis updates and feedbacks, as well as giving me with constant feedback on my work and financial support through Research Assistantship (R.A.) jobs sponsored by her grants from funding agencies such as NSERC.

Apart from my advisor, I'd like to express my gratitude to the members of my thesis committee: Dr. Dennis Borisov (my external reader), Dr. Ahmad Biniaz (my internal reader), and Dr. Boubakeur Boufama (thesis defence chair) for agreeing to serve on my thesis committee despite their busy schedules and for their insightful feedbacks and encouragement that helped me finish. I can probably agree that I have been the luckiest with this committee.

I appreciate the encouragement of my family, Sheshu and Shubha, who have given me all the support possible during this work. Finally, I would express my appreciation to all my friends (Sathya Krishna) and colleagues (Rajeswari & Manil) at the University of Windsor, for stimulating discussions, encouragement, and support throughout the duration of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER 1 : INTRODUCTION

The World Wide Web has provided a new direction in the way we communicate or administer information. With the evolving web as the information system, users are evolving with it. People are becoming increasingly enthusiastic about how the data can be obtained effortlessly within seconds from multiple resources. Kaplan & Haenlein (2010) define social media as *"A community of Internet-based apps that draw on Web 2.0's ideological and technical pillars, which allows the creation and exchange of user-generated content"*. The Web 2.0 encourages users to connect and communicate as user-generated content in a shared environment through social media discussion. People take part in reading the information and share their views on social media and other online forums like Facebook, Twitter, blogs, etc. Twitter is a popular social media platform (Microblog) on which users may voice their opinions. Twitter data opinion analysis (Alsaeedi & Zubair, 2019) is an area that has gained a great deal of interest during the last decade and includes the dissection of' tweets and the content of these phrases. ***This research focuses on the mining the summary of opinions applied to Twitter data.***

Opinion mining analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written language (Liu, 2012). Analyzing these sentiments helps customers consider other people's thoughts before using a service or buying a product. Still, it is often valuable for marketers to understand customer perceptions regarding their goods and services (Liu, 2007). Opinion mining is considered a sentiment analysis feature that provides more granular information about the product's specific feature.

To clearly understand the distinction between Opinion mining and Sentiment analysis, let us look at an example. *"Food was great. The customer service staff was unfriendly"*. Opinion mining will locate aspects in the text and their associated opinions and sentiments. Sentiment Analysis might just regard this as a negative sentence. Figure 1.1 illustrates the example and identifies the critical difference between them in opinion polarity (identification whether the Aspect is positive or negative).

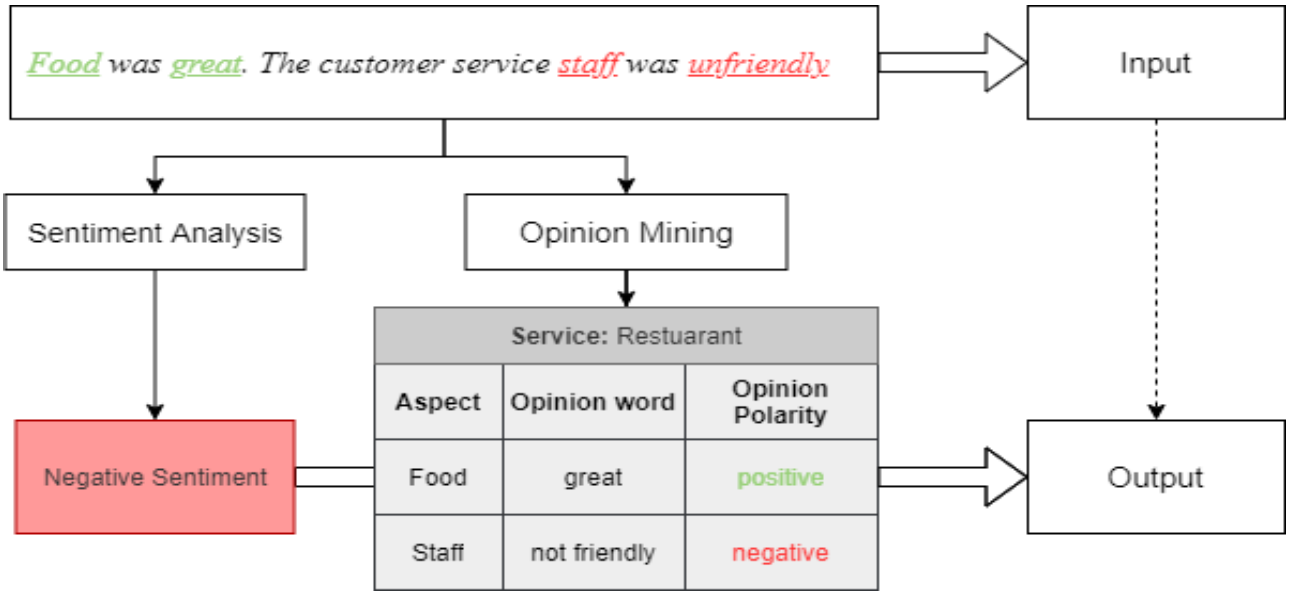

Figure 1.1: **Example of Sentiment Analysis vs. Opinion Mining**

Here, the ***food*** and ***staff*** are aspects that are attributes or components of an entity ***restaurant.*** Entities commonly refer to names of products, services, individuals, events, and organizations. Consider the following three tweets for iPhone 12 pro max:

i) *"Apple iPhone 12 Pro review: the best smartphone camera you can get https://trib.al/EX8FhLq".*

ii) *"The iphone12 pro is not cheap. #iphone12"*

iii) *"I bought an iphone12 pro max".*

In this example, one might ask what we should extract or mine from this review? The target of the opinion in this sentence is *iPhone12 Pro Max*. The target component is *"camera,"* and the opinion associated with this is *"best"*. Thus, one can determine that this review of iPhone 12 is a positive sentence or a ***positive polarity*** of the opinion. Polarity usually ranges over an ordinal scale. This scale may take the form of either an ordered set of numeric values (for example, one to five 'stars') or an ordered set of non-numeric labels (e.g., positive, negative, neutral). The only distinction between these two cases is that the distances between consecutive scores are identified in the former case. The distances are not specified in the latter one.

From the second tweet, we can see that the user has expressed that the iphone12 pro is not cheap. The target component would be *"price"*, and the opinion associated with it is *"not"*. This review can be classified into ***negative polarity.***

Neutral polarity is where people do not express any opinion. For the third tweet, the user states they have bought the iPhone without expressing any opinion about it.

1.1 Why do we need Aspect Based Opinion Mining?

The classification of text sentiments on the document and sentence level is helpful in many cases. Even so, it does not offer all the necessary information. For example, being positive about a document about a particular entity does not imply that the author's opinion is optimistic about an entity's aspects. Similarly, negative sentiments do not represent the author's negative opinion about an entity's aspects (Liu & Zhang, 2012). The classification on the document level (Moraes et al., 2013) and sentence level (Marcheggiani et al., 2014) do not provide this information. To achieve these details, we need to perform opinion mining at the aspect level (Xia et al., 2015). ***This is the primary focus of our thesis.***

1.1.1 Applications for Aspect Based Opinion Mining

i) ***Scalability*****:** It is difficult to manually go through all the reviews posted by the users to understand a specific product's specific feature. The company's performance decreases due to information overloading (exposure of too much information or data). The time gets reduced for other crucial tasks. However, Aspect based sentiment analysis does the hard work by reducing the time taken to analyze the customer's feedback.

ii) ***Competitor Analysis:*** Monitoring the product mentions online is the primary application of Aspect based sentiment analysis. Identifying the positive features would help the business to determine where the competitor is succeeding. In contrast, negative feedback given to a specific part would provide opportunities for the company.

iii) ***Target Customers:*** It helps to categorize and structure the reviews to identify the underlying patterns. Companies can then differentiate the customers as happy or unhappy customers to target them. *For example:* suppose specific customers are unhappy about a particular product or a feature of that product. In that case, companies might offer free services or promotions to make them happy.

1.2 Aspect-based opinion Mining Procedure.

Discovering all written language feelings is the prime goal of opinion mining (Saleh et al., 2011). It determines the speaker's or writer's attitude about the different aspects of a problem. We have modeled the opinion mining process in Figure 1.2, where the red color boxes indicate the focus of my thesis, in which each part has some obligations (Liu, 2012) as follows:

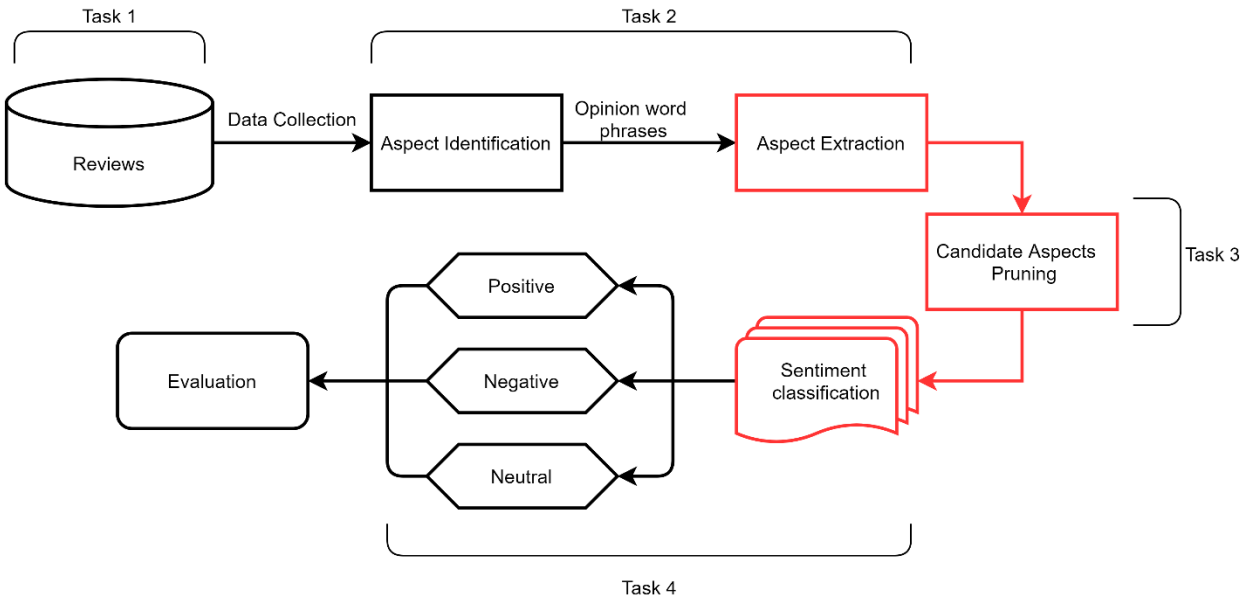

Figure 1.2: Aspect mining procedure

i) ***Task 1 (Data collection/Reviews):*** The necessary information is collected from various web resources, such as weblogs, microblogs (Twitter[1]), social networks (Facebook[2]), and review websites (Amazon[3], Yelp[4], and Tripadvisor[5]). Using tools developed to extract data through the web, and various techniques such as web scraping (Pandarachalil et al., 2015) can help collect appropriate data. In this research, Microblogs (Twitter) is considered for the generation of the review database.

ii) ***Task 2 (Aspect Identification & Extraction):*** In this phase, the frequently occurring words are selected for aspect extraction. The explanation is that frequently occurring terms are more likely to be an aspect of a product within the posts (Liu 2012); they are considered candidate aspects.

iii) ***Task 3 (Candidate Aspect Pruning):*** In this step, all the products' relevant aspects are pruned by employing data mining (explained in section 1.4) techniques.

iv) ***Task 4 (Sentiment Classification):*** Determine whether each opinion on an aspect is positive, negative, or neutral.

v) ***Task 5 (Evaluation):*** The performance of opinion classification can be evaluated using four evaluation parameters: accuracy, precision, recall, and F1-score (all discussed in Chapter 4).

1.2.1 Real-Life Application

It has become more accessible for customers to get information about specific products. It also provides excellent insight for business people to understand more about their customers.

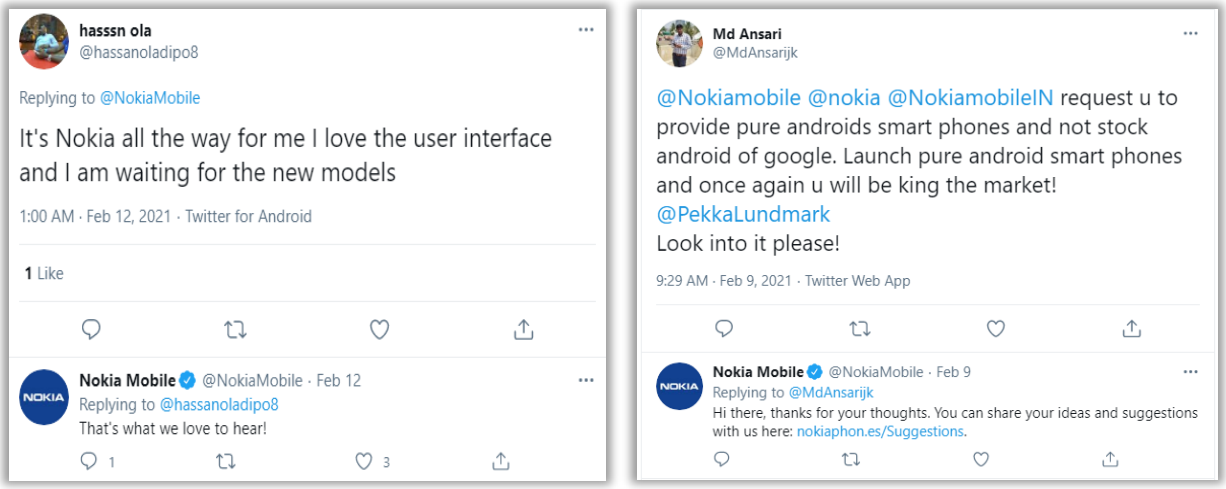

Figure 1.3: Reviews in Twitter about Nokia 8.1

1.2.2 Aspect Based Opinion Mining Terminologies

The basic terminologies currently used in aspect-based opinion mining are (Moghaddam, & Ester, 2012):

Fact: A fact is something that has occurred or is the case. Example: *"The sun is hot"*

Opinion: An opinion is a belief about subjective matters resulting from emotion or facts interpretation. Some of the keywords associated with opinions: view, think.

Subjective/Opinionated Text: Text communicating personal thoughts, opinions, or point of view, a text is subjective or opinionated, e.g., *"battery life is very good."*

Objective Text: An accurate text expresses information about the world, e.g., *"this phone lasted very long"*.

Item: An item, such as a product, service, individual, event, organization, is a concrete or abstract object. It is possible to represent an item as a hierarchy of components, sub-components, etc. A collection of one or more items is called an itemset. Example: *{bread, milk, sugar}*

Review: A review is a summary, analysis, and evaluation of a text resulting in an opinion or judgment.

Figure 1.4: A review on Twitter.

Aspect: An aspect (also called feature) is an attribute or component of the item commented on in a review. There are two types of aspects, namely:

i) **Explicit Aspects:** Aspects that are explicitly mentioned as nouns or noun phrases in a sentence, e.g., *'picture quality'* in the sentence *"The picture quality of this phone is great"*.

ii) **Implicit Aspects:** Aspects that are not explicitly mentioned in a sentence but are implied, e.g., *'price'* in the sentence *"This car is so expensive."*, or *'size'* in the sentence *"This phone will not easily fit in a pocket"*.

In the field of Aspect based opinion mining, there are two main approaches. One of them is a rule-based approach that uses natural language processing. The other is an intuitive approach that focuses on applying machine learning techniques that integrate with data mining. Many researchers have combined the methods of the two systems in a hybrid approach (monkeylearn.com).

1.3 Natural Language Processing

Natural Language Processing (NLP) is a set of computer science, information engineering, and artificial intelligence techniques for evaluating and representing naturally occurring texts. (Liddy, 2001). Opinion mining is commonly seen as a subarea of NLP and has had a considerable impact since its debut which has presented numerous new and demanding research challenges. However, research in the past fifteen years indicates that rather than being a subproblem of NLP, sentiment analysis is more like a mini version of the full NLP or a particular case of the full NLP (Liu, 2012). That is, every subproblem of NLP is also a subproblem of sentiment analysis and vice versa. The text representation process has two basic tasks, which are term indexing and term weighting. In the term indexing task, the most representative term is assigned as the document's index. In contrast, the term weighting task will give appropriate weight to the term index to measure the terms' importance throughout the document collection. Many variants of term index have been used to represent a document in the Vector Space Model (VSM) (Salton et al., 1975), such as the Bag-of-Words (BOW) (Le & Mikolov, 2014) and N-gram model (Guthrie et al., 2006; Sidorov et al., 2014).

1.3.1 Bag-of words model:

A BOW representation is an individual word unit or a unigram (1- gram) language model (Le & Mikolov, 2014) where documents are represented as a set of words that they contain along with the frequency. The steps of Bag-of-word model are as follows:

Step 1: Collect the data from the documents and separate them into each sentence in this format.

Sentence 1	it was the best of times
Sentence 2	it was the worst of times

Table 1.1: Two sentences from a book "A tale of two cities"

Step 2: Design the vocabulary

This step identifies the unique words in the table:

"it", "was", "the", "best", "of", "times", "worst"

Table 1.2: unique words in the two sentences

Step 3: Create Vectors to score the words in each document according to their sentence.

Sentence 1	it was the best of times	[1,1,1,1,1,1,0]
Sentence 2	it was the worst of times	[1,1,1,0,1,1,1,1]

Table 1.3: vector form

However, since the word order in BOW model is not preserved, it has led to semantic issues such as inaccurate representation and misleading meanings.

1.3.2 N-gram model

N-gram is a probabilistic model, which predicts or generates the next word from the previous n-1 words. The general N-gram probability estimation of the next word sequence w_n for bi-gram is denoted in for example, to approximately compute a bi-gram probability of w_n, given the previous word w_{n-1}, we will count the number of the bi-gram $C(w_{n-1}w_n)$ occurrence in the text and normalize it by dividing with the sum of unigram count for that word $C(w_{n-1})$.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

Equation 1.1: Probability of N-gram model

To understand model behaviors across varying degrees of word order distortions randomly shuffling n-grams where n = {1, 2, 3}. Shuffling 1-grams is a common technique for analyzing word-order sensitivity (Sankar et al., 2019). The ending punctuation was kept intact.

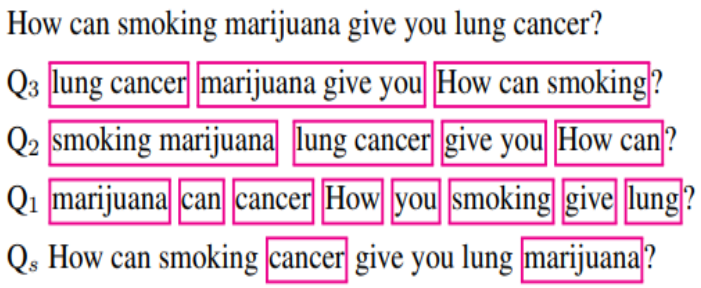

Figure 1.5: (Q3 to Q1) created by randomly shuffling 3-grams, 2-grams, and 1-grams, respectively. Qs was created by swapping two random nouns (Thang et al., 2021)

The above model even though it does consider “lung cancer” as sequence of words, it still doesn’t preserve the order. Furthermore, the model has a known issue of high dimensionality of word size combination where not all combinations are available across the collection, also known as the data sparsity issue. Now, we will see how Data mining techniques helps us to find the related aspects.

1.4 Data mining

Data mining is a technique for extracting useful knowledge from the vast collection of data according to one's business interests (Han et al., 2012). Because of the idea that "we are data rich but information poor," data mining has gotten a lot of attention for its important role in converting massive amounts of data into valuable information and knowledge. The standard data mining techniques that help in the analysis of predictions are:

1.4.1 Association Rule mining

Association rule mining looks for interesting relationships between objects in each data set. Let $I = \{i_1, i_2 \dots i_m, \}$ be a set of items. Let D, the task-relevant data, be a set of database transactions where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$. An association rule is an implication of the form $A \Rightarrow B$ where $A \subset I$, $B \subset I$, and $A \cap B = \emptyset$. The rule $A \Rightarrow B$ holds in the transaction set D with support s, where s is the percentage of transactions in D that contain $A \cup B$.

$$support(A \Rightarrow B) = \frac{number\ of\ transactions\ containing\ I}{number\ of\ transactions}$$

Equation 1.2: Support

The rule $A \Rightarrow B$ has confidence c in the transaction set D if c is the percentage of transactions in D containing A which includes B. That is,

$$confidence(A \Rightarrow B) = \frac{support\ (A\ U\ B)}{support\ (A)}$$

Equation 1.3: Confidence

Rules are considered strong that meet both a minimum support threshold (min-sup) and a minimum confidence threshold (min-conf) (Han & Kamber, 2000).

Apriori algorithm, a classic algorithm, helps mine frequent itemset and relevant association rules. It has got this odd name because it uses 'prior' knowledge of frequent itemset properties. It was first introduced by Agarwal & Srikanth (1994). To understand the associate rules better, let us look at an example.

Consider the supermarket situation in which $I = \{a, b, c, d, e\}$ is an itemset collection. There are five transactions in the database where 1 represents the object's presence and 0 represents the absence.

Transaction ID	List of items
T_1	a, b, c
T_2	b, c, d
T_3	d, e
T_4	a, b, d
T_5	a, b, c, e
T_6	a, b, c, d

Table 1.4: Transaction Database

The Apriori makes the following assumptions:

i) All subsets of the frequent itemset should be frequent.

ii) Similarly, the subsets of an infrequent itemset should be infrequent.

iii) Set a threshold support level. In our case min_sup = 50%, i.e.,

min_sup = 6 * 50% = 3.

iv) Set minimum confidence to be 75%.

Steps involved to perform the algorithm:

Step 1: Find the frequent item (L1) from the candidate set (C1).

The Apriori process's main step is to find a frequent item by counting each item's occurrence. The items that do not satisfy the minimum support count are pruned and produced frequent items (L1). In our case, frequent item (L1) = {a: 4, b:5, c:4, d:4}.

Step 2: Generate candidate set (C2) from the frequent item (L1) by Apriori join (L1 ⋈ L1).

The candidate set (C2) is generated in this step by performing L1 App-join L1. Only an item following an infrequent item in (L1) can be linked with a frequent item (L1), resulting in a candidate set (C2) = {ab, ac, ad, bc, bd, cd}.

Step 3: Find the frequent item (L2) from the candidate set (C2).

Like step 1 Frequent item (L2) is obtained by following the same procedure. The count the occurrence of each item in candidate set (C2) is calculated and infrequent items in (L1) are removed to create frequent itemset (L2) = {ab: 4, ac:3, bc:4, bd:3}.

Step 4: Generate candidate set (C_3) from the frequent item (L_2) by Apriori join ($L_2 \bowtie L_2$).

We can apply the same process as step 2 to generate a candidate set (C_3) by joining L_2 with L_2 using Apriori join. It produces candidate set (C_3) = {abc, abd, acd, bcd}.

Step 5: Find the frequent item (L_3) from the candidate set (C_3).

Here, we can see that only {abc} satisfies the minimum support threshold and is considered frequent. We stop in this step as there is no frequent itemset that meets the minimum support threshold.

To determine the Association Rules:

Rule 1: {a, b} => {c}

Confidence = support (a, b, c)/support (a, b) = 3/4 = **75% >= 75%**

Hence Rule 1 is **Selected.**

Rule 2: {b} => {a, c} means a & c – e

Confidence = support (a, b, c) / support(b) = 3/5 = **60% < 75%**

Hence Rule 2 is **not selected.**

If we set the minimum confidence to be 60%, both the rules would be considered strong.

Implementation of text data:

If we use the association rules to get the rules on the text data where each Aspect is considered as a transaction as follows:

Transaction ID	List of items
T_1	algorithm, network, graph, multicast, processor, system, parallel
T_2	cluster, network, design, message, processor, system. framework
T_3	algorithm, software, graph, method, session, analysis, parallel
T_4	switch, load, design, power, path, system, timing
T_5	cable, load, energy, power, current, motor, signal

Table 1.5: Association rule mining of text data

After the implementation of the Association rule (considering minimum support as 0.4 & confidence 1), we will get,

a. **{algorithm, graph}** => **{parallel}** from 1, 3

b. **{network, processor}** => **{system}** from 1, 2

c. **{design}** => **{system}** from 2, 4

d. **{load}** => **{power}** from 4, 5

Limitation:

Sequential ordering of events is not considered in the data analysis of association rule mining. This may contribute to the inability to identify significant trends in the details or find patterns that may not be beneficial. For example, it is often essential to understand the order of words in sentences to interpret texts (Pokou et al., 2016). The task of sequential pattern mining (Section 1.5) was suggested to solve this issue.

1.4.2 Classification

Classification is a technique that assigns categories to a collection of data. Deciding what text, word, or picture has been introduced to our senses, recognizing faces or voices, processing mail, assigning homework grades; are examples of setting an input category (Jurafsky & Martin, 2014). A general day-to-day example would be weather prediction, which uses classification to report whether the day is sunny, rainy, or cloudy. It has numerical applications ranging from target marketing, fraud detection to medical diagnosis. Decision tree induction classification (Apte & Wiess, 1997) is one of the most widely used classification techniques. The development of decision tree classifiers requires no domain knowledge or parameter setting. It is, therefore, ideal for the exploration of exploratory knowledge. A decision tree is a tree-shaped flowchart structure with a non-internal node collection (non-leaf node) denoting an attribute test. Each branch represents a test result. Each leaf node (terminal node) carries a class name. The topmost node in the tree is a root node.

For example, the following decision tree (Figure 1.6) can be used to know whether a person is eligible to get a driving license or not.

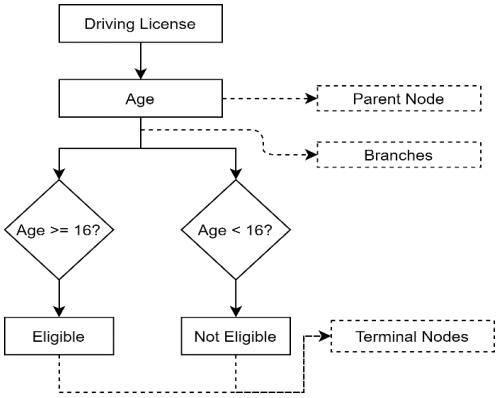

Figure 1.6: **Decision tree example**

The intuition is that if a person is below 16 years of age is not eligible.

Another widely used technique for classification is the Naïve Bayes classifier. Naive Bayes is a class of probabilistic algorithms that use Bayes' Theorem and probability theory to predict a text's tag (like a piece of news or a customer review). They're probabilistic, which means they quantify each tag's likelihood for a given text and then produce the title with the highest probability. These probabilities are calculated using the Bayes' Theorem, which determines the likelihood of a function based on prior knowledge of relevant elements.

Text	Tag
"A great game"	Sports
"The election was over."	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election."	Not sports

Table 1.6: Example for classification based on Naïve Bayes

When working with conditional probabilities Bayes' Theorem comes in handy since it allows us to reverse them:

$$P(A|B) = \frac{P\ (B|A) \times P(A)}{P\ (B)}$$

Equation 1.4: Naïve Bayes Theorem

In our case using Table 1.4, we have P (Sports | a very close game), so using this theorem, we can reverse the conditional probability:

$$P(Sports|a\ very\ close\ game) = \frac{P\ (a\ very\ close\ game|Sports) \times P(Sports)}{P\ (Sports)}$$

Equation 1.5: Naïve Bayes Theorem example for classification

1.4.3 Clustering

Clustering is the division of data into classes of items of a common type. Each category, called a cluster, consists of identical objects and is distinct from other levels of things. Clustering is like classification, except that the groups are not predefined but instead defined by the data alone. Clustering may be interpreted as partitioning or segmenting the data into classes that may or may not be disjointed, typically by evaluating the similarities between the data on predefined attributes (Dunham, 2003).

For example, let us consider:

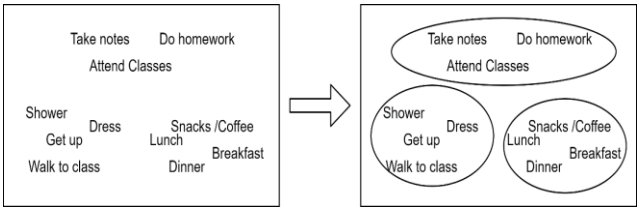

Figure 1.7: Clustering Example

The left side of the diagram gives out few words which do not wholly make any sense. If we look at the diagram's right side, a student's daily activities are listed out. So, by clustering the texts into groups, we can further gain that:

i) Group 2: 'Take notes', 'Do homework', 'Attend Classes' refers to the school work.

ii) Group 1: 'Get up', 'Shower', 'Dress', 'Walk to class's refers to the activities before school.

iii) Group 3: 'Snacks/Coffee', 'Breakfast', 'Lunch', 'Dinner' refers to eating activity.

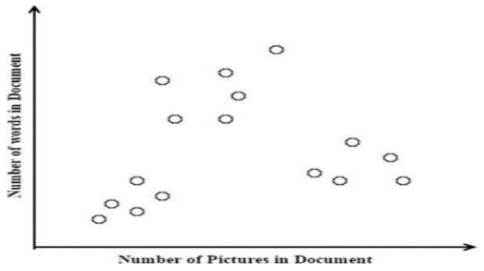

Figure 1.8: Clustering 15 Documents based on 2 Features.

Figure 1.7 is an example of the partitioning-based clustering paradigm, and the k-Means algorithm can be used for this. The steps are taken to perform k-Means go thus:

i. Choose the number of clusters (k) and the centres at random from the k locations.
ii. Assign each point to the center that is closest to form partitions.
iii. Recalculate the centre of each division after all items have been allocated to the nearest centre.
iv. Keep repeating steps ii and iii until the center stops moving.

The algorithm's objective is to minimize the objective function, E, which is:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist\,(p, c_i)^2$$

Equation 1.6: K-means Objective function

Where $\boldsymbol{p}$ represents a point and $\boldsymbol{c_i}$ is a center. In other words, the goal of k-Means is to reduce the total of the distances between each point and the center.

Hierarchical-based clustering is another clustering technique. Clustering based on hierarchical structures can be Agglomerative or Divisive. Each unit is assigned to its own cluster in Agglomerative clustering, which subsequently combines with another single cluster. This, in turn, joins with another similar cluster to form a larger cluster based on a distance measure. Clustering may be done from the bottom up.

On the other hand, Divisive clustering takes a top-bottom approach. All objects are placed in a cluster, and the cluster is broken down into smaller groups. Different clustering paradigms are density-based, graph-based, and spectral clustering (Zaki & Meira, 2014).

1.5 Sequential Pattern Mining

In data mining, two kinds of sequential data are widely used (Han et al., 2011), i.e., *time-series* and *sequences*. *Time-series* data is a collection of an ordered list of numbers. At the same time, the *sequence* is an ordered list of nominal values (symbols). For example, time-series are often used to represent the population, weight tracking, and stock prices. Sequences help us to predict the next symbol(s) based on the previously observed sequences of symbols. These symbols could represent the sequence of words in a text, a sequence of items purchased by a customer.

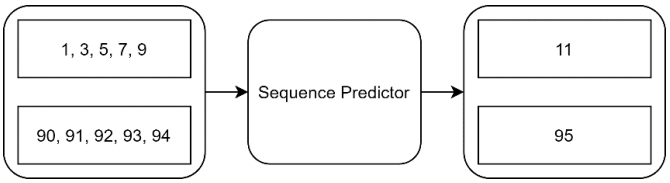

Figure 1.9: Sequential Predictor model (Jing, 2020)

A sequence pattern refers to a list of ordered events that occur concerning time and convey hidden information. Each itemset comprises sets of items separated by commas, and a sequential Pattern is typically encased between the angle brackets (>). (,). In an e-commerce system, a sequential pattern like (Bread, Milk, Tea), (Bread, Milk, Sugar, Tea), (Milk), (Tea, Sugar)> indicates that the customer bought (Bread, Milk, Tea) together in his first purchase, (Bread, Milk, Sugar, & Tea) in the second purchase, Milk alone in the third purchase, and (Tea & Sugar) together in the fourth purchase. In a sequence scenario, an item can only appear once, although in different sequence situations, it can appear multiple times. In a sequence, the number of instances of items is called the sequence length. An l-series is considered a sequence with a length of l (Han et al., 2011).

The database stores several records, where all records are sequences {s1, s2, …, and sn} arranged concerning time (Han et al., 2011) is called a sequence database. It can be represented as a tuple <SID, Sequence-item sets>, where SID: represents the sequence identifier and sequence-item sets specifies the sets in items enclosed in parenthesis ( ).

Sequence ID (SID)	Itemsets
1	CBAB
2	AACCB
3	BBAAC
4	ABACB

Table 1.7: Sequence Database example.

The sequential pattern mining task is an enumeration problem (determining the set of all solutions). It aims to enumerate all patterns (subsequences) that support no less than the user's minimum support threshold (Fournier-Viger et al., 2017). The basic solution is to test the support of all potential subsequences in a sequence database and then output just those that meet the user's set minimum support cap. Such a naive technique, however, is inefficient since the number of subsequences may be relatively high.

There have been numerous approaches proposed to cope with the problem of sequential patterns in sequence databases, and they may be divided into three groups:

i) **Horizontal database / Apriori Based:** AprioriAll (Agrawal & Srikant, 1995), GSP (Srikant & Agarwal, 1996), PSP (Masseglia et al., 1999).

ii) **Vertical database / Early pruning:** SPADE (Zaki et al., 2001), LAPIN (Yang et al., 2007), CM-SPAM (Fournier-Viger et al., 2014).

iii) **Projected database / Pattern growth:** FreeSpan (Han et al., 2000), PrefixSpan (Pei et al., 2001).

1.5.1 Why do we need Sequential patterns for feature extraction?

The problem of understanding data and its characteristics has attracted the keen interest of research from early years. The data is characterized in terms of features, also referred to as patterns or attributes. The definition of a feature is closely tied to the nature of the data. For example, for text data set, a feature can include keywords. For opinion mining, aspects and entities serve as the features. Within a pattern recognition system, feature extraction identifies features relevant to the application at hand.

In the early mining stages, words are considered as features. Such an approach is called the *bag of words* (Zhang et al., 2010) representation.

For example, let us consider the sentence:

"I bought a Nokia phone. I got my girlfriend an apple phone."

This sentence, when represented in bag-of-words, will give:
{“I”: 2, “bought”: 1, “a”: 1, “nokia”: 1, “phone”: 2, “got”: 1, “my”: 1, “girlfriend”: 1, “an”: 1, “apple”: 1}.

Although the bag-of-words model is a valuable feature extraction method, it does not consider the information embedded in words' order.

Consider the following two sentences S1 & S2 as an example:

S1: *"Only he could solve that problem"*

S2: *"He could solve only that problem"*

We cannot say that S1 is positive and S2 is negative using bag-of-word representation. It just calculates the number of words that are either present or absent in different sentences. A sequence-based model captures temporal connections between words and phrases compared to a feature focused on single words as done by the bag-of-model, which leads to the preservation of meanings of the sentences.

For example, consider the two sentences,

S3: "Cat chases a rat"

S4: "Rat chases a cat"

The sequence predictor model will capture it as S3 = ({C, R}), S4 = ({R, C}) preserving the meaning of each of the sentences, which makes it more efficient for extracting the aspects (Task 2 from section 1.2). Candidate aspect pruning (Task 3 from section 1.3) can be carried out by utilizing the minimum support threshold and the rules' confidence**. *In this thesis, we use the power of sequential patterns to leverage aspect extraction, which increases the accuracy of identifying the aspects.***

1.6 Twitter Sentiment Analysis (Thesis Motivation):

Twitter is a global social media site. When it comes to data and information, it is nothing less than a goldmine. Almost all tweets are available and readily extractable, making it possible to compile vast Twitter information for research. The fact that Twitter data is so precise makes it very good for prediction. Twitter is a microblogging service (allows users to exchange small elements of content such as short sentences, individual images, or video links) that enables users to send 'tweets' to their followers or clients. Even though a person can only create a message of 280 characters or less, this "limitation" or "feature" has not reduced users' activity. As of January 2020, Twitter has more than 340 million dynamic clients inside a given month, including 100 million clients daily. Clients' origins are widespread, with 77 % from outside the United States and sending out more than 500 million tweets every day. The Twitter site positioned thirteenth universally for activity and reacted to more than 15 billion API calls every day. Twitter information may also be seen on over a million third-party websites. Following this enormous development, Twitter has of late been the subject of much scrutiny, as Tweets frequently express client's sentiments on controversial issues. Sentiment analysis and opinion mining are complex problems in social media, owing to the massive amount of data created by humans and robots (Giachanou, & Crestani, 2016). Furthermore, we use the body of text from Twitter (Microblog) for the following reason:

i) Microblogs contain information about only one topic due to their limitation, which makes it easier for identification.

ii) Before making a purchase, consumers increasingly use social media, such as microblogs, to perform independent research. (Vollmer & Precourt, 2008).

iii) Instead of product reviews, where the product user prefers to give a one-time inspection of the product, consumers are more likely to provide updates on the performance of items over their lifespan on microblogs and in real-time.

1.6.1 Challenges of Twitter:

Twitter Sentiment Analysis (TSA) tackles analyzing the messages posted on Twitter regarding the sentiments they express. Twitter is a novel domain for SA and is very challenging. According to Giachanou & Crestani (2016), some critical challenges of studying TSA are explained in Table 1.8.

Issues	Description	Examples
Text Length	280 characters	Shooting a video today and realized 2 things. The camera on the #iPhone12Pro is 🔥 and my quarantine hair is looooooooooong.
Topic Relevance	#Hashtags	#iPhone12Pro
Incorrect English	Informal language	Looooooooong.
Data Sparsity	Misspelled words	Don't like customizing on #iphone #iPhone12Pro but atp can't beat 'em join 'em
Negation	Sarcasm detection	The design of the current #iPhone12Pro is still fundamentally the same as the original #iPhone despite the 13+ years age difference.
Stop Words	stop words like the, is, who, etc have low sentiment values	If we remove the stop words in the text length example, the text would be shooting video today realized 2 things. Camera #iphone12pro 🔥, which usually is enough to say that the camera quality is good.

Table 1.8: Challenges in Twitter Sentiment Analysis

1.7 Existing Systems:

Systems & Researchers	Research Goal	Method to Obtain Relevant aspects	Limitations
Twitter Aspect Classifier [TAC] (Lek & Poo, 2013)	Pointwise Mutual information (PMI) (Measure of association)	S1: 'Switchbot' and 'Camera' S2: 'cameras'	Does not filter the neutral statements Does not consider multi-word aspect. 'Switchbot Camera'
Microblog Aspect Miner [MAM] (Ejieh et al., 2019)	Apriori algorithm, Cosine Similarity, and K-means	S1: 'Switchbot' and 'Camera' ~~S2: 'cameras'~~	**Adv:** Filter out the neutral statements Does not consider multi–word aspect. 'Switchbot Camera'

Table 1.9: Closest existing systems based on Microblogs that considers only single word aspects

From the above Table 1.9, some of the shortcomings are common to the existing systems:

i) Many aspect expressions are multi-word phrases, which cannot be easily handled with these systems. For example, "operating system", "user manual".

ii) "life" by itself is not meaningful, whereas "battery life" is a significant aspect.

Systems & Researchers	Research Goal	Method to Obtain Relevant aspects	Limitations
Hate crime Twitter Sentiment [HCTS] (Zainuddin et al., 2018)	Association Rule mining. (Interesting relations between variables in large datasets)	"Switchbot Camera" "battery life" "life"	It does not retain the order for the classification of tweets. Redundant single aspect: It still stores "life" as a possible candidate aspect.

Table 1.10: Closest Existing Systems based on Microblogs for multi-word aspect extraction

1.8 Thesis Problem and Contributions:

i) Problem Statement:

As defined by Liu (2012), opinion is a quintuple (a set of five items),

$$(e, a, s, h, t)$$

Where *e* is the target entity, $\boldsymbol{a}$ is the targe aspect of entity $\boldsymbol{e}$ on which the opinion has been given, $\boldsymbol{s}$ is the sentiment $\boldsymbol{t}$ of the opinion on Aspect a of entity $\boldsymbol{e}, \boldsymbol{h}$ is the opinion holder, and $\boldsymbol{t}$ is the opinion posting time; $\boldsymbol{s}$ can be ***positive, negative,*** or ***neutral***. Here $\boldsymbol{e}$ and $\boldsymbol{a}$ together represent the opinion target. With the given definition, we define our problem as follows:

Given a set of microblog posts about item *P* (iPhone, Nokia), the main task is to identify P's *k* significant aspects (Single and Multi-word aspects) and to generate a summary of sentiments expressed based on the Aspect.

Multi-word Aspect Extraction: The goal of this task is to *extract aspects* of the reviewed item, and the multi-word aspect is represented as:

$$a = a_1.\ a_2 \ldots\ldots\ldots a_n$$

Equation 1.7: multi-word aspect

where a_i represents single word aspect & n represents the number of words contained in a.

Aspect-based Summary: The aim is to identify the key aspects and their polarity (positive or negative) that are being discussed in multiple reviews.

To tackle the above problem, thesis contributions are:

ii) Thesis Contributions:

A. Feature Contributions:

1) Using sequential patterns to increase the accuracy of single and multi-word aspects in microblog:

Finding the multi-word (operating system, user manual) patterns (considering that each review is a sequence of words) and retaining those words' order in Microblogs.

2) Removing the single redundant aspects that are not meaningful:

It takes input as the noun or noun phrase. It removes those nouns with no feature phrase of the superset (e.g., "battery life") to reduce the candidate generation and increase the system's accuracy.

B. Procedure Contributions:

1) The proposed approach MASM (Microblog Aspect Sequence Miner), takes nouns as input generated from the POS tags to extract the single and multi-word aspects (sequences) based on a user minimum support threshold. Instead of the Aspect Transaction Database that MAM (Microblog Aspect Miner) uses, we have generated an Aspect Sequence Database (ASD) consisting of nouns/noun phrases.
2) To remove the redundant features, we have proposed Superset Support Pruning (SSP) from the generated single word aspects and check whether no superset noun phrase does not appear together in any sentence. (e.g., manual, manual mode, manual setting).

1.9 Thesis Outline:

Chapter 2: Description of existing Text mining techniques, Aspect Extraction techniques with a new categorization based on Sequential pattern mining methods for products, and Aspect Sentiment Classification techniques. It also discusses existing systems that are based on Twitter Sentiment Analysis. Finally, a comparison of the current surveys based on those methods and the challenges of each survey it tries to solve.

Chapter 3: Discusses the proposed approach Microblog Aspect Sequence Miner to extract the aspect terms. First, we go through the preprocessing steps required to remove the special characters, URL, 'RT', '@', from the posts. It also extends by adding Slang abbreviations and emoticon abbreviations. Then we go through the Sequential pattern-based aspect extraction and sequence embedding (vector representation) of those aspects and cluster those aspects similar to the product. Finally, we rank those aspects based on the topic it belongs and generates a summary of the expressed opinions of the aspects.

Chapter 4: Discusses the experimental implementations, evaluation metrics, and proposed approach results with the existing systems.

Chapter 5: Conclusions and Future Work provides a conclusion and future work for performing Twitter sentiment analysis.

CHAPTER 2 : RELATED WORK

In this section, we address related works. We review the systems that address the text mining aspect extraction problem, aspect sentiment classification, and finally, a comparison table of the surveys related to aspect sentiment analysis.

2.1 Text Mining:

This is done as a preprocessing step before we submit the data to the Aspect Extraction phase.

Preprocessing: In preprocessing, we convert a document into a feature vector. Like considering text categorization as part of text mining, the communities have different views on how the preprocessing step should be defined. A text document often contains words that can lead to lower performance in a learning model. Terms that lead to lower performance in a learning model are often "noisy" words (Chaoji et al., 2008). Misspelled words, abbreviation words, and common words – such as "is", "or", and "a" – are often considered noise words. Such a noise word does not contain information that we can use to help in classification. We must handle these words depending on an application, and a learning algorithm referred to as text preprocessing. This thesis breaks down the preprocessing into four parts: Tokenization, Stopword Removal, Stemming, and POS tagging.

2.1.1 Tokenization:

Tokenization is the process of splitting or breaking down a vast text body into smaller lines or words. By studying the word sequence, it aids in the interpretation of the text's meaning.

For example: *"This movie is really good."*

After applying tokenization: [*'This', 'movie', 'is', 'really', 'good'*].

Tokenizers are systems that are used to tokenize data. Natural Language Toolkit tokenizer (Bird et al., 2009) is an example of a tokenizer.

2.1.2 Dropping Common terms: Stop Words

Some common terms that appear to be of little use in assisting in selecting documents that meet a user's needs are occasionally removed altogether from the lexicon (dictionary). These words are called *stop words*. The primary method for determining a stop list is to rank the terms by collection frequency (the total number of times each term appears in the document collection) and then use the most frequent words as a stop list, which is subsequently eliminated during indexing. An example of a stop list is shown in Figure 2.1.

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

Figure 2.1: **A stop list of 25 semantically non-selective words which are common (nlp.stanford.edu)**

2.1.3 Stemming:

Stemming is a crude heuristic method that removes derivational affixes off the ends of words in the hopes of reaching this aim properly most of the time.

The most common algorithm for stemming the English language, which has repeatedly been empirically very effective, is *Porter's algorithm* (Porter, 1980).

Example: Connect, Connected, Connecting, Connection, Connections

The porter Stemmer removes the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. Also, the suffix stripping process will reduce the total number of terms in the IR system and reduce the size and complexity of the system's data, which is always advantageous.

2.1.4 POS Tagging:

A Part-Of-Speech Tagger (POS Tagger) scans text in a language and assigns parts of speech to each word (and other tokens), such as nouns, verbs, adjectives, and so on. However, generally computational applications use more fine-grained POS tags like 'noun-plural'.

Example: *"Plays well with others"*

Output: *"Plays/VBZ", "well/RB", "with/IN", "others/NNS"*

For English, Penn TreeBank Tagset is the most common, and the authors claim it is 97% accurate.

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Figure 2.2: Penn TreeBank Tagset

2.2 Aspect Extraction

The problem of aspect-based Opinion mining was first studied by Hu and Liu (2004b). The idea was to extract all frequent aspects from customer reviews and then find the opinion words. Frequent aspects were those aspects about which most of the users like to express their views. Zhang & Liu (2014) classified aspect extraction methods into three categories:

i) Language Rules
ii) using supervised learning
iii) using topic models.

We extend this categorization by adding various approaches that use sequential patterns for aspect extraction, using sequential pattern mining to filter out the frequent noun phrases.

2.2.1 Language Rules

Frequency-based approaches typically apply a set of constraints to phrases with high-frequency nouns to define aspects. A noun, adjective, verb, or adverb may convey an aspect. People are more likely to speak about relevant aspects in comments, indicating which aspects should be collective nouns. However, not all the frequent nouns are aspects. Therefore, different filtering techniques are applied to frequent nouns to filter out non-aspects.

FBS: Mining and summarizing customer reviews (Hu & Liu, 2004)

Hu and Liu (2004) introduced the FBS approach, which mines product characteristics from customer evaluations, detects sentiment opinion, and summarises the explicitly expressed elements' outcomes. This paper, FBS, serves as the base for introducing the world of Aspect based opinion mining as it lays out the tasks described in Section 1.2 (Aspect based opinion mining procedure). The input to FBS is a product name and an entry web page for all the product reviews. FBS method has the following tasks:

1.2 Parts of Speech tagging:

The NLProcessor linguistic parser is used to parse each review to split the sentences into text. Each sentence that has been tagged is kept in the review database.

Example: *"I am absolutely in awe of this camera"*

('I', 'PRP'), ('am', 'VBP'), ('absolutely', 'RB'), ('in', 'IN'), ('awe', 'NN'), ('of', 'IN'), ('this', 'DT'), ('camera', 'NN').

For the POS tag description, refer to Figure 2.2.

1.3 Frequent Feature Identification:

Association miner CBA (Liu et al., 1998), which is based on the Apriori algorithm (Agarwal & Srikant, 1994) with minimum support of 1%, is applied to obtain the frequently occurring nouns or noun phrases that are explicitly mentioned in the reviews. The generated frequent itemsets are also called candidate aspects.

Example: Assuming there are three sentences in the review, the frequently occurring nouns are shown below:

Sentence #	Noun/Noun Phrase
1	camera, the focus, manual, a broad strap
2	the memory card, lens,
3	bright pictures, camera, zoom

Table 2.1: Sample Structure of Transaction File

1.4 Feature Pruning:

The input to this stage is those candidate aspects generated by CBA. However, not all candidate aspects are genuine, and two pruning types are employed to remove those unlikely features.

Compactness Pruning – checks if the features contain at least two words, which are called *feature phrases*. For example, 'digital camera' is not compact in Sentence 3 below but compact in Sentence 1 and Sentence 2.

Sentence 1: "I constantly searched for a ***digital camera*** for more than three months."

Sentence 2: "This is the best ***digital camera*** on the market."

Sentence 3: "The ***camera*** does not have a ***digital*** zoom."

Redundancy Pruning - removes candidate aspects that contain single words. For example, *life* is not a helpful feature, while *battery life* is a meaningful feature phrase.

1.5 Opinion Word extraction:

Opinion words are mainly used to communicate personal feelings. Previous research on subjectivity has found a positive, statistically significant relationship between adjectives with subjectivity (modifiers).

Opinion sentence: A statement is called an opinion sentence if it contains more product characteristics and one or more opinion words.

Effective opinion: Nearby (closest) adjective is recorded as its effective opinion, for each feature in a sentence

Example: *horrible* is the effective opinion of the strap in *"The strap is really horrible and obstructs the way of parts of the camera all the time."* Effective opinions will be useful when predicting the orientation of opinion sentences.

1.6 Opinion Summarization:

According to the opinion sentence orientations, related opinion sentences are classified into positive and negative categories for each discovered feature. A count is determined to illustrate how many reviewers offer positive/negative views of the feature. The output is a summary of the reviews as shown below:

Digital_camera_1:
Feature: **picture quality**
Positive: 253
Negative: 6
Feature: *size*
Positive: 134
Negative: 10
…

Figure 2.3: Opinion Summarization example (Hu & Liu, 2004)

Limitations: This method tends to develop too many non-aspects and neglect low-frequency aspects. Also, they require multiple parameters (thresholds) to be manually calibrated, making it impossible to port them to another dataset.

OPINE: Extracting Product Features and Opinions from Reviews (Popescu & Etzioni, 2005)

Popescu & Etzioni (2005), like Hu & Liu (2004), first extracted all nouns from reviews and retained those with a frequency greater than an experimentally set threshold. The difference is in evaluating the candidate feature (Task 3) that uses the Pointwise Mutual Information (PMI) assessment. The calculation of PMI (Turney, 2001) is computed between each fact and automatically generated discriminator phrases (e.g., "great X", "has X", "comes with X" where X

is the product aspect). Given a noun phrase f and discriminator d, the PMI score is defined as follows:

$$PMI\ (f, d)\ =\ \frac{Hits\ (d + f)}{Hits\ (d) \times Hits\ (f)}$$

Equation 2.1: Point-Wise Mutual information

$Hits$ refers to the number of aspects returned.

Example: If a google search for "iPhone" (target product) returns 10 results, a search for "camera" (an aspect of target entity) returns 20 results, and a search for "iPhone AND camera" returns 5 results, the PMI value of "iPhone" and "camera" is calculated as follows:

$$PMI\ (iPhone, camera)\ =\ log_2 \frac{5}{10\ \times\ 20}\ =\ -2.60$$

OPINE applies an NLP parser to determine syntactic dependencies of words in each sentence and then generates a set of syntactic rules for extracting sentiment associated with each Aspect. Finally, a classification technique is applied to the extracted sentiments to classify them as positive or negative. The precision of the OPINE saw a significant rise of 22% compared to the FBS system.

Red Opal: Product Featuring scoring from reviews (Scaffidi et al., 2007)

The method proposed by Scaffidi et al. (2007) compares the frequency of extracted candidates (frequent noun phrases) in a review corpus with their occurrence rates in generic English. This work is a follow-up on Hu & Liu (2004). It improves the latter by using baseline statistics of words in English and probability-based heuristics to identify product categories. Before aspect extraction, Red Opal is provided with statistics on lemma frequencies in generic text.

Opinion Zoom: (Marrese- Taylor et al., 2013a, 2013b, 2014)

Marrese-Taylor et al. (2013b) proposed an extension to aspect-based opinion mining techniques for the tourism domain, i.e., hotels and restaurants. They define a sentence as an ordered set of tokens, and tokens could be words or punctuations. If any token comes twice in a sentence, it will be considered two separate tokens at distinct positions. By this definition, they calculated

the distance between two words and used this to extract aspects. Further, they have followed rule-based techniques (Ding et al. 2008) to determine the opinion orientation. This technique was also adopted to propose OpinionZoom (Marrese-Taylor et al. 2013a), modular software to evaluate the tourism domain's opinions. Moreover, Marrese-Taylor et al. (2014) extended the same work to develop a generic architecture to create a prototype that analyzed the tourism domain's opinions from tripadvisor.com. Their methods showed a low precision for explicit aspect extraction, i.e., 35%, but F-measure was 92% for sentiment orientation.

2.2.1.1 Summary of the systems based on Linguistic Rules (Frequency and Relation based)

Name	Description	Limitations
FBS (Hu & Liu, 2004)	Proposed a system that mines and summarizes all the customer reviews of a product. It assumes that frequent nouns/noun phrases are the aspects of a product. Then, an orientation identification algorithm based on a pre-defined seed set (e.g., a small group of opinion words) and WordNet's semantic structure is employed to identify the opinion orientation automatically.	This method tends to produce too many non-aspects and miss low-frequency aspects.
OPINE (Popescu & Etzioni, 2005)	Employs an aspect assessment method based on pointwise mutation information (PMI) and the syntactic dependency rules to improve the quality of extracted aspect terms and opinion expressions.	They require the manual tuning of various parameters (thresholds), making them hard to port to another dataset.
Red Opal (Scaffidi et al., 2007)	The proposed approach firstly detects the frequent uni-gram nouns and noun phrases. Then, a 100-million-word corpus is employed as the general corpus to evaluate aspect candidates. High score candidates are considered as aspects.	Minimal test sets of reviews are used, resulting in poor evaluation of the system.
Opinion Digger (Moghaddam & Ester, 2010)	Uses the known aspects from reviews to extract explicit aspects and was responsible for ranking from 1 – 5 based on rating guidelines.	this method fails to handle similar syntactic structures and therefore cannot be generalized for unseen data

Table 2.2: Comparison of Existing Systems based on Language Rules.

2.2.2 Extraction using supervised learning:

Aspect extraction is a particular case of the general information extraction problem. Many algorithms based on supervised learning have been proposed in the past for information extraction.

In aspect-based opinion mining, these methods can be applied to reviews to identify Aspect, sentiments, and polarity. The most prominent methods for information extraction are based on sequential learning (or sequential labeling). The current state-of-the-art sequential learning methods are HMM (Hidden Markov Model) (Rabiner, 1989) and CRF (Conditional Random Field) (Lafferty et al., 2001).

i) Hidden Markov Model:

Assume you're trapped in a room for several days with no access to the outside world. You want to forecast the weather outside, but the only information you have is whether the person who brings your daily food into the room is carrying an umbrella.

Weather	Probability of Umbrella
Sunny	0.1
Rainy	0.8
Cloudy	0.3

Table 2.3: Probability $P(x_i|q_i)$ of carrying an umbrella ($x_i = true$) based on the weather q_i on some day i.

However, the actual weather is still hidden, and we want to find the probability of a particular weather $q_i \in Sunny, Rainy, Cloudy$ can only be based on the observation x_i(umbrella). This conditional probability $P(q_i|x_i)$ can be re written according to Bayes' rule:

$$P(q_i|x_i) = \frac{P(x_i|q_i)P(q_i)}{P(x_i)}$$

Equation 2.2: Bayes Rule

or, for n days, and weather sequence $Q = \{q_1, \ldots, q_n\}$, as well as 'umbrella $sequence' X = \{x_1, , \ldots, x_n\}$.

$$P(q_1, , \ldots q_n|x_1, , \ldots x_n) = \frac{P(x_1, , \ldots x_n|q_1, , \ldots q_n)P(q_1, , \ldots q_n)}{P(x_1, , \ldots x_n)}$$

The probability of $P(x_1, , \ldots x_n|q_1, , \ldots q_n)$ can be assumed:

$$\prod_{i=1}^{n} P(x_i|q_i)$$

if we assume that, for all i, the q_i , x_i are independent of all x_j and q_j, for all $j \neq i$.

We want to conclude the weather outside based on our assumptions (whether the individual has an umbrella). Thus, we omit the probability of seeing an umbrella $P(x1, \ldots, xn)$ as it

is independent of the weather, that we like to predict. We get a probability measure proportional to the probability, which we can call the likelihood L.

$$P(q_1, , \dots q_n | x_1, , \dots x_n) \; \alpha$$
$$L(q_1, , \dots q_n | x_1, , \dots x_n) \quad = P(x_1, , \dots x_n | q_1, , \dots q_n) \, . P(q_1, , \dots q_n)$$

It can be re written in the form:

$$P(q_1, , \dots q_n | x_1, , \dots x_n) \; \alpha$$
$$L(q_1, , \dots q_n | x_1, , \dots x_n) \quad = \prod_{i=1}^{n} P(x_i | q_i) \prod_{i=1}^{n} P(q_i | q_{i-1})$$

ii) Conditional Random Field:

One limitation of the HMM is that its assumptions may not be adequate for real-life problems, which leads to reduced accuracy. To address the limitation, linear chain CRF (Lafferty et al., 2001) was proposed as an undirected sequence model, which models the conditional distribution $p(y \mid x)$ over hidden sequence y given observation sequence x (Sutton & McCallum, 2011). The conditional model is trained to label an unknown observation sequence x by selecting the hidden sequence y that maximizes p (y | x). The model thereby allows the relaxation of HMM's strong assumptions of independence. The linear-chain CRF model is illustrated in Figure 6.3, where.

$$y = < y_1, y_2, \dots, y_t > : hidden\ state\ sequence$$
$$x = < x_1, x_2, \dots, x_t > : observation\ sequence$$

The conditional distribution p (y | x) takes the form.

$$p\,(y|x) = \frac{1}{Z(x)} \prod_{i=1}^{t} \exp \left\{ \sum_{k=1}^{k} \lambda_k \, f_k(y_i, y_{i-1}, X_i) \right\}$$

Equation 2.3: Conditional distribution

where Z(x) is a normalization function

$$Z(x) = \sum_{y} \prod_{i=1}^{t} \exp \left\{ \sum_{k=1}^{k} \lambda_k \, f_k(y_i, y_{i-1}, X_i) \right\}$$

Equation 2.4: Normalization function

CRF introduces the concept of feature functions. Each feature function has the form $f_k(y_i, y_{i-1}, X_i)$ and λ_k is its corresponding weight. CRF makes independence assumption among y, but not among x. One argument for the feature function f_k is the vector X_iThis means each feature function can depend on observation x from any step. Thus, CRF can introduce more features than HMM at each stage (Sutton & McCallum, 2011).

OpinionMiner (Jin et al., 2009)

The model proposed by Jin et al. (2009), called OpinionMiner, is based on HMM. The main tasks of this model are identifying aspects, sentiments, and their polarity. The novelty of this work is integrating POS information with the lexicalization technique. In other words, the model integrates POS information in the HMM framework, i.e., the generation of each word depends not only on its previous term but also on its part of the speech tag.

To simplify the approach and make it computable, three assumptions have been made:

i) The current tag t_i depends on the previous tag t_{i-1} and the word w_{i-1};
ii) The probability of a current word w_i only depends on the current POS tag si and the previous work w_{i-1};
iii) The probability of current POS s_i depends on the current tag t_i and the previous work w_{i-1}. Based on these approximations, model parameters could be estimated by maximum likelihood estimation when given an annotated training corpus.

Example: '*I love how easy it is to transfer the pictures to my laptop.*'' could be tagged as ''<BG> *I* </BG> <OPINION_POS_EXP> *love* < BG> how< /BG> < PROD_FEAT>*easy it is transfer the pictures*< /PROD_FEAT> < BG> *to*< /BG> < BG> *my* < /BG> < BG > laptop< /BG>.'' Thus, the aspect '*ease of transferring the picture*' and opinion '*love*' could be extracted.

Skip Tree CRF (Li et al., 2010)

Li et al. (2010) proposes a series of CRF models for extracting aspects, related sentiments, and the polarity of sentiments from reviews. Besides the neighbor context modeled by linear-chain CRF, they propose using Skip-chain CRF and Tree CRF to utilize the sentence's conjunction structure and syntactic tree structure. The Skip-chain CRF model assumes that if words or phrases are

connected by the conjunction 'and', they mostly have the same opinion polarity. It makes the reverse assumption for words joined by the conjunction 'but'. Tree-chain CRF considers the syntactic tree structure of reviews, which provide deeper syntactic dependencies for aspects and sentiments. A unified model, called Skip-Tree CRF, is proposed to integrate these two structures.

> **Example:** *"iPhoneX has a great camera and a cool appearance"*, two long-distance dependencies *(dep (great, cool) and dep (camera, appearance))* could be captured by the Skip-Tree CRF.

2.2.2.1 Comparison of aspect extraction techniques based on Topic modeling.

Name	Description	Limitations
Opinion Miner (Jin et al., 2009)	A hybrid approach integrating POS information with the lexicalization technique under the HMM framework. In this model, the current tag is related to the previous title and correlates to prior observations (word token and part of speech).	The requirement of manually labelled data for training which is not readily available.
Skip-Tree CRF (Li et al., 2010)	Incorporates the syntactic tree structure into the CRF framework and outperforms traditional CRF	Did not cover other type of elements such as intensifiers, phrases, and infrequent entities.
L-CRF (Shu et al., 2017)	Incorporated lifelong learning into CRF and the proposed approach performs markedly better than the traditional CRF.	The systems do not consider the previous extraction results and the knowledge in the last CRF models.

Table 2.4: Comparison of Existing Systems based on Supervised Learning

2.2.3 Extraction using Topic models:

Topic modeling is an unsupervised technique in machine learning capable of scanning a collection of documents, identifying clusters of terms and phrase patterns within them, and automatically clustering word classes and related expressions that better describe a set of documents (monkeylearn.com). There is no need for manually labeled data compared to sequential models, as discussed in the section. Also, the topic model performs both aspect extraction and grouping simultaneously in an unsupervised manner. It assumes that every sentiment is a blend of different topics and each topic under discussion is a probability distribution of different words.

Example: The rating labels as usual 'pos' and 'neg' have been used.

Inputs: Review set of Nokia phones.

Outputs: <Nokia, sound, pos>,

<Nokia, price, neg>,

<Nokia, screen, pos>,

<Nokia, weight, neg> and….

From this example, the Aspect-based opinion mining model's output gives useful information about public opinion on *'Nokia'* and more granular data about *'sound'*, *'price'*, *'screen'*, and *'weight'*.

Advantages:

1) The extracted aspects can be grouped according to semantic similarity (metric defined over a set of documents).

   Example: "Car" is related to "bus" but is also related to "road" and "driving".

2) The proposed approaches are domain-independent and could be transferred to a new domain easily.

Hence topics from the models can be considered as aspects. Topic modeling can thus be applied to extract aspects. However, there is also a difference, i.e., topics can cover both parts and sentiments. For aspect-based opinion mining, they need to be separated. This separation can be achieved by extending the basic topic models to model both aspects and sentiments jointly. There are two main basic topic models: Probabilistic Latent Semantic Indexing (PLSI) () and Latent Dirichlet Allocation (LDA).

Probabilistic Latent Semantic Indexing:

PLSI is a statistical approach for analysing data with multiple co-occurrences. PLSI is often called the aspect model (). The four tasks of the PLSI model are:

i. Associates an unobserved latent class variable $z \epsilon \mathbb{Z} = \{z_1, \ldots, z_K\}$ with each observation.
ii. Defines a joint probability model over documents and words.
iii. Assumes w is independent of d conditioned on z.
iv. The cardinality of z should be much less than d and w.

The model formulation of PLSI is given below:

i) Basic Generative model

a) Select document d with probability $P(d)$

b) Select a latent class z with probability $P(z|d)$

c) Generate a word w with probability $P(w|z)$

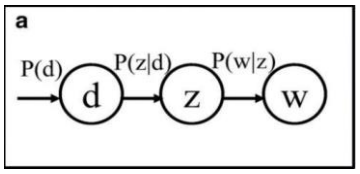

Figure 2.4: Basic Generative model

ii) Joint Probability model

$$P(d,w) = P(d)P(w|d) \qquad P(w|d) = \sum_{z \in \mathbb{Z}} p(w|z)\,P(z|d)$$

Equation 2.5: Joint probability model

Example:

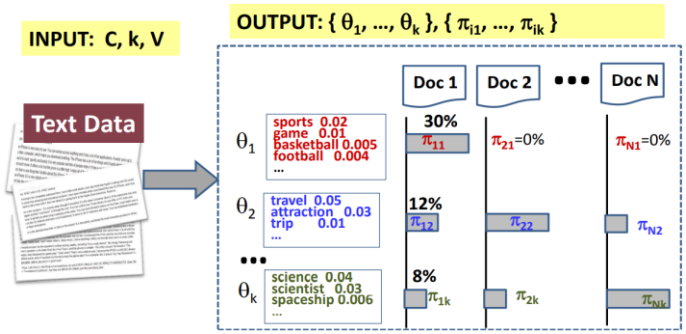

Figure 2.5: Mining Multiple topics from text using PLSI.

Problems of PLSI:

- There's no clear way to use it to assign a probability to a word that hasn't been encountered before.
- The linear increase of parameters indicates that the model is prone to overfitting, which is a serious problem experimentally.

LDA Latent Dirichlet Allocation:

LDA is like PLSI, except that in LDA, the topic distribution is assumed to have a Dirichlet prior, resulting in more good mixtures of topics document. In both models, aspects are represented as mixtures over latent topics associated with a distribution of vocabulary words.

A k-dimensional Dirichlet random variable θ can take values in the $(k-1)-$ simplex (a k-vector θ lies in the $(k-1)-$simplex if $\theta_i \geq 0$, $\sum_{i=1}^{k} \theta i = 1$) and has the following probability density on this simplex:

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_i - 1} \ldots . \theta_k^{\alpha_k - 1},$$

Equation 2.6: Probability density

where the parameter α is a k-vector with components $\alpha_i > 0$ and $\Gamma(x)$ is the Gamma function.

The Generative Process:

LDA assumes the following generative process for each document w in a corpus D:

1. Choose $N \sim Poisson\ \xi$.
2. Choose $\theta \sim Dir\ \alpha$.
3. For each of the N words wN:
   a. Choose a topic $zn \sim Multinomial\ \theta$.
   b. Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Example: Assume you have a collection of articles that can be classified into two groups, each defined by the parameters shown below:

a) **Animals:** tiger, lion, fox, wolf, monkey.
b) **Politics:** Democrat, Republic, Congress.

2.2.2 Comparison of Existing Systems based on the topic model.

Name	Description	Limitations
ORMFW (Khalid et al., 2018)	The proposed approach makes use of linguistic associations to identify prominent aspect terms for an aspect.	It works better on limited data domains.

SA-ASM (Amplayo et al., 2018)	Focuses on improving the aspect term extraction of topic models by incorporating product descriptions to the current state-of-the-art sentiment topic model, Aspect Sentiment Unification Model (ASUM)	difficulty in capturing aspects related to different domains, as well as emotion-related data

Table 2.5: Comparison of Existing Systems based on Topic Model.

2.3 Existing Sequential pattern algorithms

2.3.1 GSP: Generalized Sequential pattern algorithm by (Srikant & Agarwal, 1996)

Problem Statement: To find the entire set of sequential patterns in the database using GSP, given a sequence database S (Table 2.6) and the min support threshold.

SID	Sequence
1	< (A, B), (C), (F, G), (G), (E) >
2	< (A, D), (C), (B), (A, B, E, F) >
3	< (A), (B), (F, G), (E) >
4	< (B), (F, G) >

Table 2.6: Sequence Database

Input is Sequence database, min_support = 2, candidate set (C1) = {A, B, C, D, E, F, G}.

Output: Frequent Sequential patterns.

Step 1: Find 1- frequent sequence (L1) that satisfies minimum support.

L1 = {<(A): 3>, <(B): 4>, <(C):2>, <(E):3>, <(F):4>, <(G): 3>}. Even though (A) has appeared multiple times in the first sequence, only its support is counted, i.e., whether it has appeared or not. Therefore, the support for (A) is 1 in SID (1). Also. (D) is filtered out since its support is 1 in the whole set and does not satisfy the minimum support.

Step 2: Using L1 *GSPjoin* L1, generate a candidate sequence (Ck=2). To produce a bigger candidate set 2, link the 1-frequent sequence (L1) from step 1 using the GSP join method, which may be expressed as L (k-1) *GSPjoin* L. (k-1). If subsequences generated by removing the first element of W1 and the last element of W2 are the same, every sequence (W1) discovered in the first L (k-1) must connect with another sequence (W2) in the second. In this example, we're making sequences with candidate 2 (Ck=2), which can provide = $\binom{6}{2} = 6 \times 6 + \frac{6 \times 5}{2} = 51$ candidates

(Similar to picking a team of 2 people in a group of 6). Therefore C2 = <(A), (A)>, <(A), (B)>, …… <(B), (A)>, …. <(G), (G)>, < (A, B)>, …. < (F, G)>.

Step 3: To maintain the only sequence with an occurrence or support count in the database more than or equal to the minimal support, find 2-frequent sequences (L2) by counting the occurrence of 2-sequences in candidate sequence (C2). For example, L2 = < (A, B)>, <(A), (C)>, <(A), (E)>, <(A), (F)>, <(A), (G)>, <(B), (E)>, <(B), (F)>, <(B), (G)>, (C), (E)>, <(C), (F)>, <(F), (E)>, <(F, G)> , <(G), (E)>.

Step 4: Repeat the candidate generation and pruning procedure until the collection of frequent sequences generated (Ck) and pruned (Lk) is empty.

Output: Frequent sequences as a union of L1 U L2 U L3 U L4 U … L_k.

1 – Frequent	2 – Frequent	3 – Frequent	4 – Frequent
<(A)>, <(B>, <(C)>, <(E)>, <(F)>, <(G)>	< (A, B)>, <(A), (B)>, <(A)>, <(A), (E)>, <(A), (F)>, <(A), (G)>, <(B), (E)>, <(B), (F)>, <(B), (G)>, (C), (E)>, <(C), (F)>, <(F), (E)>, < (F, G)>, <(G), (E)>	<(A), (B), (E)>, <(A), (B), (F)>, <(A), (C), (E)>, <(A), (C), (F)>, <(A), (F), (E)>, <(A), (F, G)>, <(A), (G), (E)>, <(B), (F), (E)>, <(B), (F, G)>, <(B), (G), (E)>, < (F, G), (E)>	<(A), (F, G), (E)>, <(B), (F, G), (E)>

Table 2.7: Frequent Sequences Table

Limitations of GSP:

i) **Multiple scans of databases.**

Every database search increases the length of each candidate sequence by one. Example: To find <(abc) (abc) (abc) (abc) (abc)>, GSP must scan the database at least 15 times.

ii) **Difficulties at mining long sequential patterns.**

There is only a single sequence of length 100, min_sup = 1

Length – 1 candidate sequences = 100

Length – 2 candidate sequences = $100 \times 100 + \frac{100 \times 99}{2} = 14950$

Length - 3 candidate sequences = $\binom{100}{3} = 161700$

Total = $\sum_{i=1}^{100} \binom{100}{i} = 2^{100} - 1 \approx 10^{30}$

2.3.2 Prefix Span: (Prefix-projected sequential pattern mining) algorithm by (Pei et al., 2001).

Input: sequence database (Table 2.8), **Min. support** = 2**, Candidate sets** = {A, B, C, D, E, F},

Output: Frequent sequential patterns.

SID	Sequences
100	<(A), (A, B), (A, C), (D), (C, F)>
200	< (A, D), (C), (B), (A, E)>
300	< (E, F), (A, B), (D), (C), (B)>
400	<(E), (G), (A, F), (C), (B), (C)>

Table 2.8: Sequence Database

Step 1: Count the number of supporters for each unique item and preserve only sequences with a support count more than or equal to the minimum of two, as shown in Table 2.9.

<(A)>	<(B)>	<(C)>	<(D)>	<(E)>	<(F)>	<(G)>
3	4	2	1	3	4	3

Table 2.9: Support of Singleton Sequence

Step 2: Prune singleton sequences that have a specified minimum threshold. In our case, the minimum support is 2, and we need to prune <(D)> since it does not satisfy minimum support.

Step 3: A projected database is created by considering a 1-frequent sequence from a sequential database. For example, the projected database of frequent 1 sequence <(A)> would consist of all the items that appear after the sequence <(A)> (will consist of all the sequences with the prefix <(A)>). The projected database for all of the items with support 1 is shown in Table 2.6.

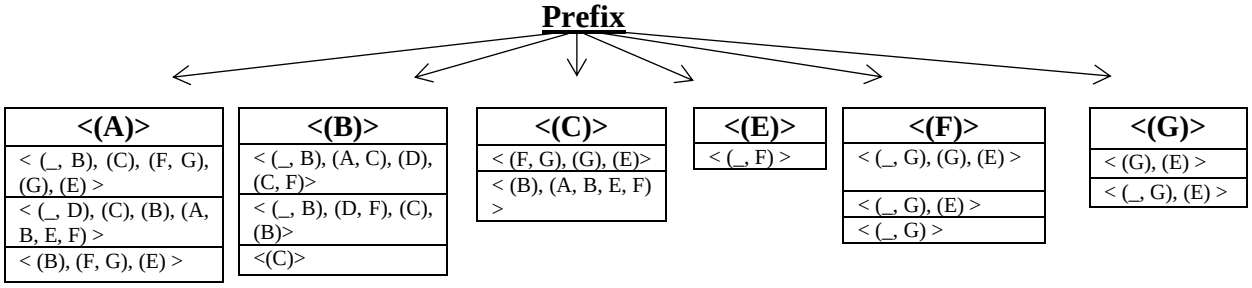

Table 2.10: Projected Database

Step 4: Find the frequent sequences from the projected databases and repeatedly check with the minimum threshold until no projected database is created.

- Find the sequence present in the projected database. Let us consider the projected database of

  <(C)> is present in Table 2.7.

<(C)>
< (F, G), (G), (E)>
< (B), (A, B, E, F) >

Table 2.11: Projected Database for <C>

- The predicted database is searched for the most common items. Only <(B)> and <(C)> are common in our case.

<(A)>	<(B)>	<(C)>	<(D)>	<(E)>	<(F)>	<(G)>
0	1	0	0	2	2	1

Table 2.12 Frequent Items in Project Database

- Now, the projected database for sequence < (C), (E)> and < (C), (F)> are constructed using step 4. Furthermore, their respective projected databases are examined for the most often occurring items.

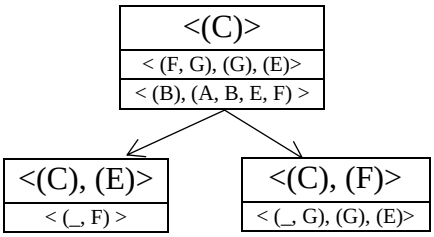

Table 2.13: Projected Database for Sequence <(C), (E)> and <(C), (F)>

Step 5: <(C), (E)> and <(C), (F)> is infrequent, i.e., does not satisfy the minimum support threshold, the process will get terminated. The same is done for all the other steps.

Limitations of Prefix Span:

- it can be costly to repeatedly scan the database and create database projections in terms of runtime.
- Creating database projections can consume a considerable amount of memory if it is naively implemented. In the worst case, it requires copying almost the whole database for each database projection.

2.3.3 Sequential Rule mining:

Sequential Rule mining (Fournier-Viger et al., 2014; Forunier-Viger et al., 2017) is a variant of sequential pattern mining in which sequential rules of the form X Y are discovered, indicating that if some items X appear in a sequence, they will be followed by some other things Y with a given confidence. The concept of a sequential rule is like that of association rules. Sequential rules must be mined in sequences rather than transactions, and X must occur before Y according to the sequential ordering. Sequential rules identify an important limitation of sequential pattern mining: although some sequential patterns may frequently appear in a sequence database, the patterns may have very low confidence and thus be worthless for decision-making or prediction.

For example, consider the Table 2.14,

SID	Sequence
1	< {a, b}, {c}, {f, g}, {g}, {e}>
2	< {a, d}, {c}, {b}, {a, b, e, f}>
3	<{a}, {b}, {f}, {e}>
4	<{b}, {f, g}>

Table 2.14: Sequence Database

The sequential pattern <(f)(e)> is considered frequent if min_sup = 2 because this pattern appears in 2 sequences. Thus, it may be appealing to think that f is likely to be followed by e in other sequences. However, this is not the case. By looking at Table 2.14, it can be found that f is actually followed by e in only two of the four sequences where f appears. This example shows that sequential patterns can be misleading.

Sequential rules address this problem by considering not only their support but also their confidence. For example, the sequential rule {f} → {e} has a support of 2 sequences and a confidence of 50%, indicating that although this rule is frequent, it is not a strong rule. Formally, the confidence of a sequential rule X → Y is defined as the number of sequences containing the items X before the items Y divided by the number of sequences containing the items X (Fournier-Viger et al., 2017). Numerous sequential rule mining has been proposed, such as Rule Growth (Fournier-Viger et al., 2017) and ERMiner (Fournier-Viger et al., 2014), which adopt a pattern-growth and a vertical approach for discovering rules.

2.3.4 Comparison of the existing Systems that utilize Sequential patterns for AE.

Paper	Published Year	Method	Cons
Opinion Feature Extraction Using Class Sequential Rules	2006	Class Sequential Rules + Prefix Span	Not reviewed on full texts.
Aspect-based opinion mining from product reviews	2012	GSP (generalized Sequential Pattern)	GSP is slow, and it can find sequences that appear many times
Exploiting Sequential Patterns to Detect Objective Aspects from Online Reviews	2016	Prefix Span	Only worked on Objective aspects.
Sequential patterns rule-based approach for opinion target extraction from customer reviews	2018	Prefix span + Sequential Rules	Repetitive patterns

Table 2.15 Comparison of Existing Systems based on SPM for Aspect Extraction.

2.4 Aspect Based Sentiment Classification:

Sentiment classification is a special type of non-topic-based text categorization. The predefined classes are the overall sentiments of the aspects. For example, the positive and negative sentiments are used as class labels for the movie reviews dataset (Pang et al., 2002). Sentiment classification has been used for customer review analysis and summarizing opinions on web pages, such as a newsgroup, forum, and blog (Li & Zong, 2008). The most famous text classification examples are opinion mining (Sentiment Analysis) and topic labeling (understanding what a given text is talking about: positive or negative).

The most common classifiers that are used for Opinion Mining are:

2.4.1 Supervised Learning

a) Support Vector Machine:

Support Vector Machine (SVM) is the best classifier that provides the most accurate speech classification problems. They achieved this by creating a hyperplane with maximal Euclidean distance for the nearest trained examples. The hyperplane of the Support Vector Machine is completely resolved by a small subset of the prepared data sets that are regarded

as support vectors. The qualified classifier does not have access to the remaining training datasets. So, the classifier SVMs have been applied successfully and used in different sequence processing applications for text classification. SVMs have been used in hypertext and text classification since they do not require labeled training data set.

b) Neural Network:

The neural network includes numerous neurons in which this neuron is its fundamental unit. Multilayer neural networks were used with non-linear margins. The results of the neuron in the previous layer will be given as input for the next layer. This type of classifier training of the data set is more complicated because the faults must be backpropagated for various layers.

c) Naïve Bayes: A Naive Bayesian classifier is one of the familiar supervised learning techniques frequently used for classification purposes. Their classifier is naive since it considers the contingencies that are linked are not depending on the further. The substance in combining all the single word feasibility reports in the file would be the calculation of overall document feasibility. Because they have less computational power than other approaches, these Nave Bayesian classifiers have been widely used in sentiment categorization, however independence assumptions will lead to incorrect findings.

d) Maximum entropy: The Maximum Entropy classifier is defined by a weight set that is utilised to associate with the collective future, which is achieved by encoding a training data set. Because its work is done by generating certain data sets against the input and binding them directly, this Maximum Entropy classifier occurs alongside classifiers like log-linear and exponential classifiers. The exponent of the result will be used.

e) Decision Tree: In the Decision Tree classifier, the interior nodes were marked with features and edges, leaving the node named a trial on the data set weight. Leaves in the tree are called categorization. This categorizes the whole document by starting at the tree's root and moving successfully through its branches until a leaf node is reached. Learning in a decision tree adopts a decision tree classifier as an anticipated model. It maps information of an item to conclusions of that item’s expected value. A large amount of input can be figured out using authoritative computing assets in finite time in a decision tree. The decision tree classifier's primary advantages are that it is simple to comprehend and interpret. This classifier requires

small data preparation. But these concepts can create complicated trees that do not make generalized easily.

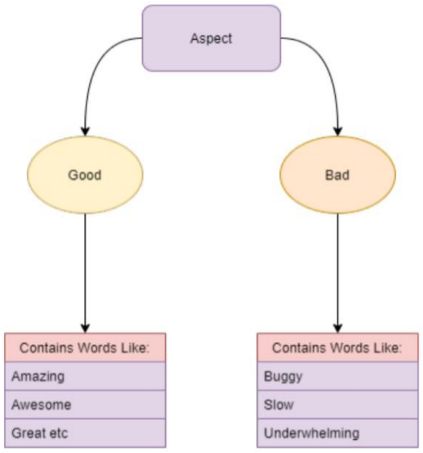

Figure 2.6: **Decision Tree For a sentiment towards an Aspect**

2.4.2 Lexicon Based:

a) Dictionary-based approach.

In this approach, a set of opinion words are manually collected, and a seed list is prepared. Then we search for dictionaries and thesaurus to find synonyms and antonyms of text. The newly found synonyms are added to the seed list. This process continues until no new words are found.

Disadvantage: difficulty in finding context or domain-oriented opinion words

b) Corpus-based approach

Corpus is a collection of writings, often on a specific topic. In this approach, a seed list is prepared and is expanded with the help of corpus text (Keshtkar & Inkpen, 2012). Thus, it solves the problem of the limited domain-oriented text. It can be done in two ways.

Statistical approach: This approach is used to find cooccurrence words in the corpus. The idea is that if the word appears mostly in positive text, then its polarity is positive. If it mainly occurs in the negative text, then its polarity is negative.

Semantic approach: This approach calculates sentiment values by using the principle of similarity between words. Wordnet can be used for this purpose. Synonyms and antonyms

of given words (Medhat et al., 2014) can be found using this, and sentiment value can be calculated.

2.5 Existing Systems that perform ABOM in Microblogs (Twitter):

Twitter Aspect Classifier (TAC):

Input: microblog post

Output: corresponding list of possible aspect candidate terms along with their opinions and polarity

Step 1: Aspect-sentiment extraction (input and pre-processing of data)

Given a tweet, this step determines a list of possible aspect candidates. The authors utilised a Parts-of-Speech (POS) tagger (pennbank tree tagset), a sentiment lexicon (SentiWordNet), and gazetteer lists (a stop word list, a swear word list and an intensifier word list). The POS tagger is used to give tags to each word in the tweet (for example, N for Nouns and V for Verbs). Each noun is regarded as a potential aspect word. The emotion is then determined by the closest verb to the left of the noun. They look up the polarity of the verb in the sentiment lexicon, which becomes the sentiment polarity of the aspect word.

Example: Consider the following Twitter posts:

Post 1: #Technews How to Set Up Android Wear for iPhone http://t.co/JQKQa8PmKE.

Post 2: LOT of 140 iPhone 5/C/S Cracked screens with GOOD LCD TESTED! - Full read by eBay: Priceâ€¦

The task is to mine the iPhone's aspects from these collections of microblog posts and determine the opinion polarity on each aspect. Table 1 shows the possible candidate aspect.

SN	Possible Candidate Aspects	Posts in which they occur
1	#Technews	post 1
2	Android	post 1
3	Wear	post 1
4	Screens	post 2
5	eBay	post 2
6	LCD	post 2

Table 2.16: Possible Candidate Aspects

Step 2: The aspect candidates acquired in the previous stage are replaced with the adjective, verb, or adverb (referred to as modifier) that is closest to the left of the aspect candidates obtained in the previous step.

For example, in P1, the nearest modifier (adjective, verb, or adverb) to the aspect candidate, "Android," is "set". So set is chosen, and the polarity of the set (positive or negative) is checked in a lexicon (SentiWordNet). It gives us a results list (Table 2.17) containing the possible aspect candidate, the nearest modifier, and its polarity.

SN	Possible Candidate Aspects	Left Hand Modifier	Polarity of Modifier
1	#Technews	NA	Neutral
2	Iphone	set	Neutral
3	Wear	set	Neutral
4	Screens	cracked	Negative
5	eBay	read	Neutral
6	LCD	good	Positive

Table 2.17: Result list from Step 2

Step 3: Aspect Pruning Stage.

The aspect candidates acquired in the previous stage are replaced with the adjective, verb, or adverb (referred to as modifier) that is closest to the left of the aspect candidates obtained in the previous step. The formula for PMI is given by (Turney 2001):

$$PMI\ (p, q)\ = log_2 \frac{Hits\ (P\ AND\ q)}{Hits\ (p) \times Hits(q)}$$

Equation 2.7: Pointwise mutual information as defined in TAC

p – the product (for example, "iPhone"), q – a product feature (for example, "Camera"), Hits(p) is the number of results returned by a search for the word "p."

SN	q	Hits of q	Hits p AND q	PMI
1	#Technews	185,000	47,500	-32.355
2	Iphone	1,380,000,000	796,000,000	-31.187
3	Wear	947,000,000	99,100,000	-33.649
4	Screens	594,000,000	39,500,000	-34.304
5	eBay	286,000,000	54,100,000	-32.795
6	LCD	348,000,000	90,200,000	-32.341

Table 2.18: Pointwise Mutual Information

Step 4: Assuming we want to get the top 5 aspects of the product, we obtain the highest PMI scores. For this example, the principal 5 aspects are Android, Wear, Screens, LCD, and #Technews. The opinion polarity of these characteristics is determined by checking the lookup database.

Finally, the **output** will be:

SN	Aspect	Polarity
1	Iphone	Neutral
2	Wear	Neutral
3	Screens	Negative
4	LCD	Positive
5	#Technews	NA

Table 2.19: Polarity based on the aspect.

Microblog Aspect Miner (Ejieh et al., 2019)

In this paper, the authors have increased the accuracy of extracting aspects in the presence of spam posts, advertisements, buzz posts, competitor's products.

Input: The product name, e, is used as a Twitter API search query.

Output: A sorted list of the most important features of the Product e that are being discussed on Twitter.

Step 1: This step of MAM will remove the foreign characters and URL and stop words from generating cleaned posts. It obtains the subjective posts by running the preprocessed posts through the subjectivity post computation algorithm.

Example: Consider the following Twitter posts:

i. *@Android i cant connect my iphone 6 with the android moto 360. Help me please*
ii. *Definitely have to get this iPhone screen fixed!!*

After preprocessing and running the subjectivity module.

i. i cant connect my iphone 6 with the android moto 360. Help me please.
ii. Definitely have to get this iPhone screen fixed

Step 2: Then it tokenizes the subjective posts to obtain:

i. 'cant', 'connect', 'iphone', '6', 'android', 'samsung', 'galaxy', 'watch', '.', 'help', 'please',
ii. definitely', 'have', 'get', 'iphone', 'screen', 'fixed'

Step 3: The part-of-speech tags (POS Tags) are applied to each of the word tokens in the subjective postings, and the Nouns and Plural Nouns are selected.

‘iphone’:3, ‘phone’:2, ‘help’:1.

Step 4: In this step, prune off the list of nouns by selecting only nouns that occur with minimum support of 1% in the subjective posts as our frequent nouns. Some of the semantic similarity between each frequent noun and the entity are: (help:0.3306), (iPhone: 1.0000), (screen: 0.5685), (periscope: -0.0737). Our frequent noun list (words that represents the aspects) becomes: {battery, back, lol, iPhone, get, screen, phone, cases, Android, charger}.

Step 5: K-Means clustering algorithm is applied to this pruned frequent noun list to divide them into two clusters:

Cluster 1	{get, back, lol}
Cluster 2	{android, cases, iPhone, phone, screen, battery, charger}.

Cluster 2 is selected because it has the entity term (iPhone in this case) as the candidate aspects.

Step 6: The authors develop a concept called the Aspect-Product Similarity Threshold to get the relevant aspects (APSM). This is the point at which the cosine similarity between a product and its aspect goes below a certain threshold. This threshold has been determined to be 0.7 in experiments. Above this threshold, the most likely candidates are competitors' products or the product's parent company. As a result, they are not considered to be part of the development process. The cosine similarity is also used to prioritise the important characteristics. The higher a candidate aspect's classification as a product aspect is, the closer it is to the APSM.

The terms iPhone, Android, and Phone having an APST score greater than 0.7 are removed. As a result, the Aspect Mining Module returns the following as aspects of the entity, iPhone, with APST scores below 0.7: screen, charger, battery, and cases. These are graded by how closely they resemble the iPhone.

Step 7: Using the discovered aspects, the next step is to get people's opinions on each of these discovered aspects to know if they are positive, negative, or neutral by running them through the Aspect Opinion Mining (AOM) module. Then refer to the subjective posts to get the posts in which these discovered aspects were mentioned, following a summary of each aspect's opinions as to the final output—SN Frequent Nouns Similarity with Entity.

1. cases Negative (100%); Positive (0%); Neutral (0%)

2. screen Negative (100%); Positive (0%); Neutral (0%)

3. battery Negative (100%); Positive (0%); Neutral (0%)

4. charger Negative (100%); Positive (0%); Neutral (0%)

2.6 Comparison of Existing Surveys

2.6.1 Comparison of Surveys Referred for Aspect Based Sentiment Analysis:

Author	Title	Idea	Challenges
Liu (2012)	Sentiment Analysis:	• covered the field of SA at document, sentence, and aspect-level. • discussed various issues related to Aspect Extraction, sentiment classification, sentiment lexicons, Natural Language Processing, and opinion-spam detection. • surveyed the till date practical solutions along with the future directions	How to cope with review ranking, redundancy issues, viewpoints quality, genuine aspects, spammer detection, etc.…?
Ravi et al. (2015)	A Survey on Opinion Mining and Sentiment Analysis: Tasks, Approaches, and Applications	• organized subtasks of machine learning, NLP, and SA techniques, such as subjectivity classification, sentiment classification, lexicon relation, opinion-word extraction, and various applications of SA • discussed open issues and future directions in SA	How to focus on sentence-level and document-level SA and their subtasks?
Schouten et al. (2016)	Survey on Aspect-Level Sentiment Analysis	• performed approach-based categorization of different solutions related to AE, aspect classification, and a combination of both. • proposed future research direction for semantically-rich-concept-centric AbSA.	How to cope with the challenges of comparative opinions, conditional sentences, negation modifiers, and presentation?
Nazir et al. (2020)	Issues and Challenges of Aspect-based Sentiment Analysis: A Comprehensive Survey	• discusses the problems and challenges of AE, ASA, and SE • presents the progress of AbSA by concisely describing the recent solutions. • highlight factors responsible for SE dynamicity.	• How to improve the mechanism of AE? • What measures should be taken to achieve good classification accuracy at the aspect level?

		• proposes future research directions by critically analyzing the present solutions.	• How to predict SE dynamicity?

Table 2.20: Comparison of Surveys Referred for Aspect Based Sentiment Analysis.

2.6.2 Comparison of Surveys referred for Sequential Pattern Algorithms:

Author	Title	Idea	Challenges
Mabroukeh & Ezeife, (2010)	A Taxonomy of Sequential Pattern Mining Algorithms	• provides a comparative performance analysis of many critical techniques and discusses theoretical aspects of the taxonomy categories.	What are the essential features that a reliable sequential pattern-mining algorithm should provide?
Fournier-Viger et al. (2018)	A survey of Sequential pattern algorithms	• give an overview of sequential pattern mining as well as a review of current developments and future research directions	What are the most recent techniques, advances, and challenges in the field of Sequential pattern mining?

Table 2.21: Comparison of Surveys referred for Sequential Pattern Algorithms.

2.6.3 Comparison of Surveys referred for Deep Learning-based Sentiment Analysis:

Author	Title	Idea	Challenges
Zhang et al. (2018)	Deep Learning for SA: A Survey	• presented applications and deep-learning approaches for the SA related tasks such as sentiment intersubjectivity, lexicon expansion, stance detection	How to achieve advances in SA using deep learning approaches?
Do et al. (2019)	Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review	• The goal of this research is to talk about and compare recent developments in DL approaches in general, as well as Aspect Based Sentiment Analysis.	How to investigate deep neural networks as well as recent trends in research in Aspect Based Sentiment Analysis?

Table 2.22: Comparison of Surveys referred for Deep Learning-based Sentiment Analysis.

2.6.4 Comparison of Surveys referred for Twitter Sentiment Analysis:

Author	Title	Idea	Challenges
Giachanou & Crestani, (2016)	Like It or Not: A Survey of Twitter SA Methods	• discussed the deep-learning algorithms related to Twitter SA • elaborated tasks specific to emotion detection, change of sentiment over time, sarcasm detection, and sentiment classification	How to tackle the challenges, tasks, and feature selection methods limited to Twitter SA?
Zimbra et al. (2018)	The State-of-the-Art in Twitter SA: A Review and Benchmark Evaluation	• focused on challenges and key trends related to classification errors, Twitter monitoring, and event detection to perform twitter SA effectively	How to reveal the root causes of commonly occurring classification errors?

Table 2.23: Comparison of Surveys referred for Twitter Sentiment Analysis.

CHAPTER 3 : PROPOSED SOLUTION

The main algorithm for determining the sentiment of the expressed aspects is shown below in Figure 3.1.

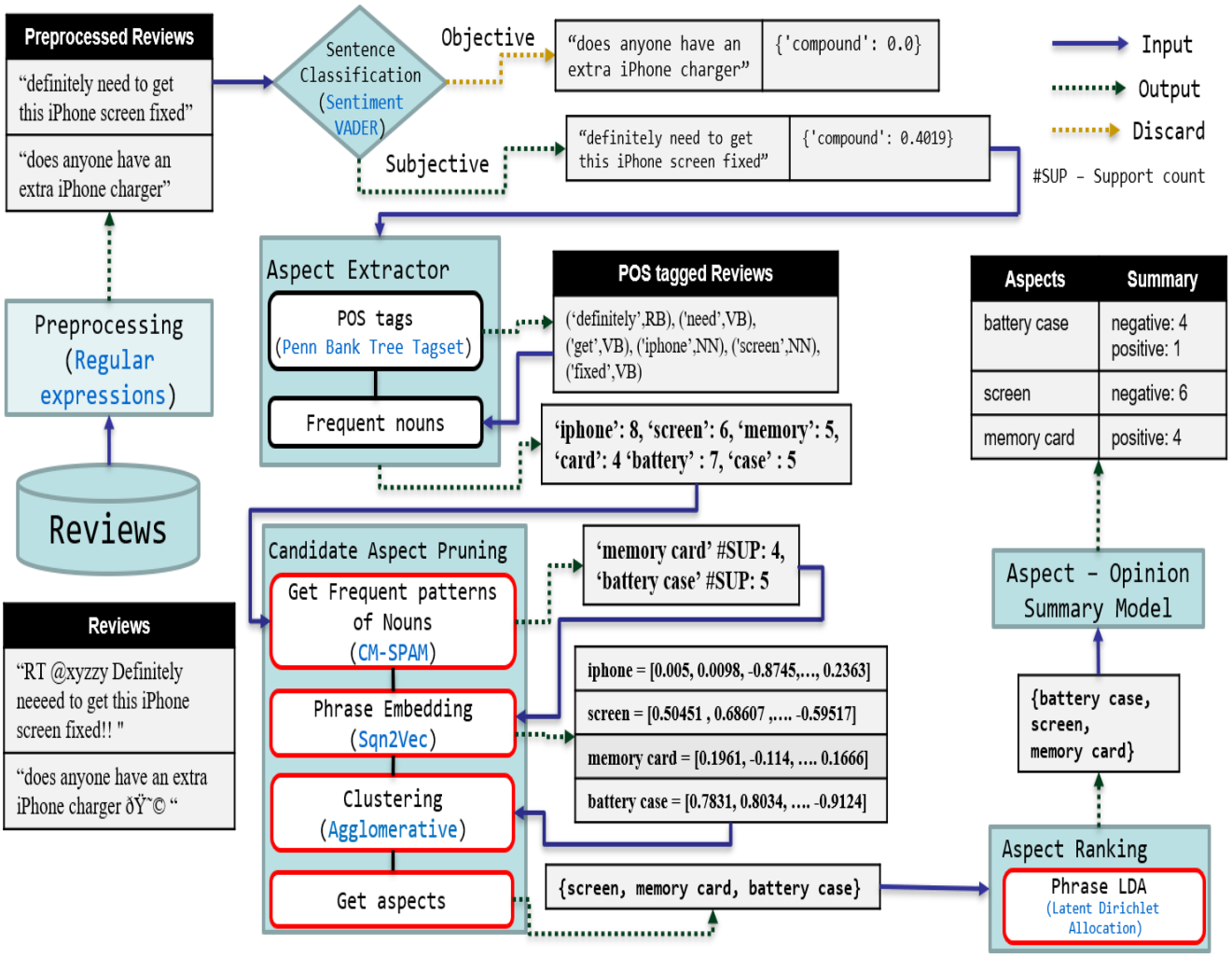

Figure 3.1: The proposed Architecture

3.1: Preprocessing of Tweets:

Input: Twitter comments or Text data

Output: Pre-processed Text data for next step of Natural Language Pre-processing Task.

For each **review** in **Twitter Data File**

Initialize temporary empty string processed Tweet to store the result of output.

1. Replace all URLs or https:// links with the word 'URL' using regular expression methods and store the result in processed Tweet.
2. Replace all '@username' with the word 'AT_USER' and store the result in processed Tweet.
3. Filter All #Hashtags and RT from the comment and store the result in processed Tweet.
4. Look for repetitions of two or more characters and replace them with the character itself. Store result in processed Tweet.
5. Filter all additional special characters (: \ | []; : {} - + ( ) < > ? ! @ # % *,) from the comment. Store result in processed Tweet.
6. Remove the word 'URL', which was replaced in step 1, and store the result in processed Tweet.
7. Remove the word 'AT_USER', which was replaced in step 2, and store the processed Tweet result.

Return processed Tweet.

Step 1: 1 All URLs in the tweets are removed by the algorithm.

	reviewText
Input Tweet	WOOOW!!! Great news. @Samsung comes with a monstrous 108mp camera giving tough competition to high-end mobiles like @googlepixel and @iphone RT The new Samsung galaxy series provides a massive 108mp rear camera. http://t.co/O3wZGPsAxx. #camera #samsung #galaxy #note
URL Processed Tweet	WOOOW!!! Great news. @Samsung comes with a monstrous 108mp camera giving tough competition to high-end mobiles like @googlepixel and @iphone RT The new Samsung galaxy series provides a massive 108mp rear camera. **URL** #camera #samsung #galaxy #note

Table 3.1: URL Processed Tweet

Step 2: The '@username' is removed from the tweet.

	reviewText
URL Processed Tweet	WOOOW!!! Great news. @Samsung comes with a monstrous 108mp camera giving tough competition to high-end mobiles like @googlepixel and @iphone RT The new Samsung galaxy series provides a massive 108mp rear camera. **URL** #camera #samsung #galaxy #note
@Username Processed Tweet	WOOOW!!! Great news. **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** RT The new Samsung galaxy series provides a massive 108mp rear camera. **URL** #camera #samsung #galaxy #note

Table 3.2: @username replaced with AT_USER

Step 3: A retweet occurs when a person re-post a comment on another user's account, describing the user's reaction to that specific post (Hemalatha et al. 2012). In the current step retweets are removed along with the event tags that are the information after the "#hashtags." It might have sentimental value. As a result, we just removed the symbol '#' to retain the meaning of the term.

	reviewText
@Username Processed Tweet	WOOOW!!! Great news. @Samsung comes with a monstrous 108mp camera giving tough competition to high-end mobiles like @googlepixel and @iphone RT The new Samsung galaxy series provides a massive 108mp rear camera. **URL** #camera #samsung #galaxy #note
RT Processed Tweet & #hashtags removed	WOOOW!!! Great news. **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** The new Samsung galaxy series provides a massive 108mp rear camera. **URL** #camera #samsung #galaxy #note

Table 3.3: "RT" removed tweets

Step 4: People frequently use the word with multiple characters to show their strong sentiments (Hemalatha et al., 2012). For example, 'YEEEESSS". The number of 'Es' in this word is excessive and does not belong in lexical resources (), thus it must be removed.

	reviewText
RT Processed Tweet & #hashtags removed	WOOOW!!! Great news. **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** The new Samsung galaxy series provides a massive 108mp rear camera. **URL** #camera #samsung #galaxy #note
Repetitive words Processed Tweet	WOW!!! Great news. **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** The new Samsung galaxy series provides a massive 108mp rear camera. **URL** camera samsung galaxy note

Table 3.4: Elongated words

Step 5: Unnecessary whitespaces at the start, middle, and end of tweets, special characters like punctuation, and character repetition may also be found in user-generated data. To begin, all excess white space was removed using Python's built-in function. Second, all the tweets' meaningless and needless special characters were removed (Hemalatha et al., 2012). These characters include: \ | [ ] ;: {} - + ( ) < > ? ! @ # % *, and a few more. These characters have no distinct and unique meaning, and they don't specify whether they're employed for positive or negative. As a result, the best choice is to get rid of them. Also, these special characters are occasionally added to words like "Great!"" A dictionary would not contain words with special characters (in this example, an exclamation mark (!)) if you compared these terms. As a result, the dictionary would be unable to locate the corresponding meaning. If the comment was positive but the dictionary didn't identify the term, the polarity of the positive comment would be reduced, turning it into a neutral comment with incorrect results.

	reviewText
Repetitive words Processed Tweet	WOW!!! Great news. **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** The new Samsung galaxy series provides a massive 108mp rear camera. **URL** camera samsung galaxy note
Special Character Processed Tweet	WOW Great news **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** The new Samsung galaxy series provides a massive 108mp rear camera **URL** camera samsung galaxy note

Table 3.5: Punctuations and Whitespaces

Step 6: Remove the term URL from the comment and save the result.

	reviewText
Special Character Processed Tweet	WOW Great news **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** The new Samsung galaxy series provides a massive 108mp rear camera **URL** camera samsung galaxy note
URL Removed Tweets	WOW Great news **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** The new Samsung galaxy series provides a massive 108mp rear camera camera samsung galaxy note

Table 3.6: URL removed tweets

Step 7: Remove the term AT USER from the comment and save the result.

	reviewText
Special Character Processed Tweet	WOW Great news **AT_USER** comes with a monstrous 108mp camera giving tough competition to high-end mobiles like **AT_USER** and **AT_USER** The new Samsung galaxy series provides a massive 108mp rear camera camera samsung galaxy note
AT_USER tweets removed	WOW Great news comes with a monstrous 108mp camera giving tough competition to high-end mobiles like and the new Samsung galaxy series provides a massive 108mp rear camera camera samsung galaxy note

Table 3.7: Final preprocessed tweets

3.2 Subjectivity Module

In this step, we will get the overall sentiment of the sentence. This approach relies on the use of a lexicon. A lexicon is a collection of entries containing information on words (or word stems); information about a word can include its part(s) of speech, spelling variants, inflectional variants, encoded syntactical information, and so on. There are several sentiment lexicons available that are designed particularly for sentiment analysis. This level of analysis is close to subjectivity classification (Wiebe, Bruce, and O'Hara, 1999), which distinguishes sentences (called objective sentences) that express factual information from sentences (called subjective sentences) that express subjective views and opinions. In this step, we will get the overall sentiment of the sentence and we will pass positive and neutral opinions for feature extraction.

3.2.1 VADER (Valence Aware Dictionary for sEntiment Reasoning)

VADER (Hutto & Gilbert, 2014) is a lexicon and rule-based sentiment analysis method built for analyzing sentiment from social media with more than 9000 lexical words. Vader combines sentiment lexicons (i.e., list of lexical words) and sentence characteristics (semantic orientation of words) to determine a sentence polarity. The Positive, Negative and Neutral scores represent the proportion of text that falls in these categories.

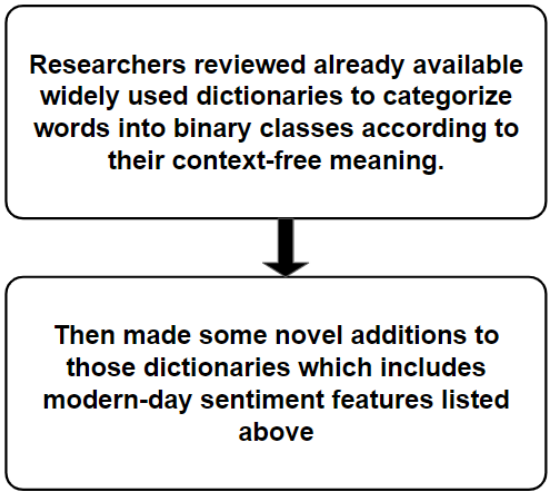

Figure 3.2: Two step process of creating a lexicon dictionary

Using a 'Wisdom of the Crowd' (WotC) method, VADER researchers verified the broad application of these lexical characteristics responsible for emotions.

WotC is based on the notion that a collection of people's collective knowledge, as conveyed through their aggregated views, may be trusted as a substitute for expert knowledge. This allowed them to obtain a reliable point estimate for each context-free text's sentiment valence score.

Mturk (Amazon Mechanical Turk) is a well-known crowdsourcing marketplace where distant expert raters undertake activities such as rating speeches.

Valence score of some context-free texts is:

- Positive Valence score: "okay" is 0.9 "good" is 1.9, and "great" is 3.1,
- Negative Valence score: "horrible" is –2.5 and emoticon ' :(' is –2.2.

3.2.2 Calculation of Valence Scores

Heuristics are rules that VADER uses to include the influence of each subtext on the perceived strength of feeling in sentence-level text. There are a total five of them. These heuristics go beyond what a conventional bag-of-words model would typically capture. They include connections between words that are affected by word order.

Five Heuristics are explained below: -

i) **Punctuation,** namely the exclamation mark (!) raises the intensity without changing the meaning direction. “The weather is cool!!!,” for example, is more intense than “The weather is cool.”

ii) **Capitalization**, especially the use of ALL-CAPS to emphasise a sentiment-relevant term in the context of other non-capitalized words, enhances sentiment intensity without altering semantic direction. For example: “The weather is COOL.” conveys more intensity than “The weather is cool.”

iii) **Degree modifiers** (also known as intensifiers, booster words, or degree adverbs) change the intensity of a feeling by raising or reducing it. For example: “The weather is extremely cold.” is more intense than “The weather is cold.”, whereas “The weather is slightly cold.” reduces the intensity.

iv) **Conjunctions cause polarity shifts;** the contrastive conjunction "but" indicates a shift in sentiment polarity, with the sentiment of the paragraph after the conjunction taking precedence. For example: “The weather is cold, but it is bearable.” has mixed sentiment, with the latter half dictating the overall rating.

v) **Catching Polarity Negation**, we identify almost 90% of situations when negation reverses the polarity of the text by looking at the continuous sequence of three items before a sentiment-laden lexical characteristic. For example, a sentence negated would be “The weather isn't really that cold.”.

For example: “The iPhone is super cool”. Our sentence was rated as 67% Positive, 33% Neutral and 0% Negative.

In our case, lexicon ratings for each word in VADER is “super (2.9) and cool (1.3)” =x (4.2).

Sentiment Metric	Score
Positive	0.674
Neutral	0.326
Negative	0.0

Table 3.8: Valence scores of Sentiment VADER

3.2.3 Reason for selecting Compound score values for sentiment classification

The compound score is calculated by adding the valence ratings of each word in the lexicon, adjusting them according to the criteria, and then normalising them to a range of -1 (most extreme negative) to +1 (most extreme positive) (most extreme positive). This is the ideal metric to use if you want a single unidimensional measure of emotion for a certain text.

where x = sum of valence scores, and α = Normalization constant (default value is 15)

$$x = \frac{x}{\sqrt{x^2 + \propto}}$$

Equation 3.1: Calculation of compound scores

So, for the above example,

$$x = \frac{4.2}{\sqrt{4.2^2 + 15}} = \frac{4.2}{5.71} = 0.735.$$

3.2.4 Reason for selecting VADER over other sentiment lexicons

Some of the advantages of Sentiment VADER are:

i) It does not require any training data.
ii) It can very well understand the sentiment of a text containing emoticons, slang, conjunctions, capital words, punctuations and much more.
iii) It works well on social media text.
iv) VADER can work with multiple domains.

The below shows that the accuracy obtained by the Sentiment VADER in comparison to other existing systems (Al-Shabibi, 2020) is higher for the Sanders corpus twitter dataset (Sanders, 2011).

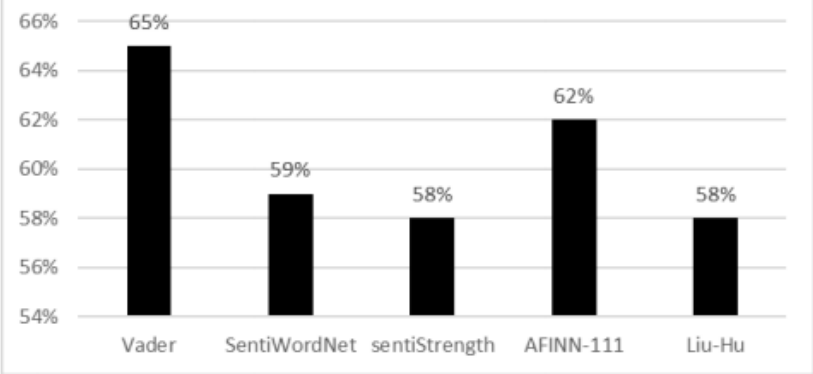

Figure 3.3: Accuracy obtained in comparison of VADER (Al-Shabi, 2020)

Sentiment Lexicon	accuracy	Positive			Negative			Neutral		
		P	R	F	P	R	F	P	R	F
Vader	65%	0.49	0.72	0.59	0.67	0.53	0.59	0.84	0.53	0.65
SentiWordNet	59%	0.43	0.43	0.45	0.46	0.42	0.47	0.74	0.71	0.73
sentiStrength	58%	0.29	0.63	0.4	0.45	0.49	0.47	0.82	0.59	0.68
AFINN-111	62%	0.3	0.53	0.38	0.51	0.41	0.45	0.78	0.68	0.73
Liu-Hu	58%	0.3	0.59	0.4	0.44	0.49	0.46	0.79	0.6	0.68
P: precision R: recall F: F1-measure										

Table 3.9: Evaluation criteria in the Sanders dataset (Al-Shabi, 2020)

3.3 Frequent Noun/Noun Phrase Identification:

The MASM applies CM-SPAM algorithm on the remaining nouns to find all multi-part noun phrases which are frequent, e.g., photo quality and LCD display. We modify the algorithm so that the position of words in the sentences are considered. This would lead to frequent phrase mining.

For example, let us consider a review:

“The camera quality is bad”.

After POS tagging:

[(‘The’, ‘DT’), (‘camera’, ‘NNP’), (‘quality’, ‘NNP’), (‘is’, ‘VBZ’), (‘bad’, ‘JJ’)].

So, mining semantically meaningful phrases has the following advantages:

i) Change the granularity of text data from words to phrases.

ii) Improve the power and efficiency of unstructured data manipulation.

For example, consider a text dataset with two sentences:

iii) S1 = *"machine learning is a field of computer science".*

iv) S2 = *"machine learning gives computer systems the ability to learn".*

SP	Symbols	Sup
X_1	{a}	1.00
X_2	{g}	1.00
X_3	{t}	0.75
X_4	{a, g}	0.75
X_5	{g, a}	0.75

Seq	SPs
S_1	$\{X_1, X_2, X_3, X_4, X_5\}$
S_2	$\{X_1, X_2, X_3, X_4, X_5\}$
S_3	$\{X_1, X_2, X_3, X_5\}$
S_4	$\{X_1, X_2, X_4\}$

Seq	Symbols
S_1	{c, a, g, a, a, g, t}
S_2	{t, g, a, c, a, g}
S_3	{g, a, a, t}
S_4	{a, g}

Although two Sequential Patterns X1 = {*machine, learning*} and X2 = {*machine, computer*} are found in both S1 and S2, X2 is less meaningful than X1 due to the large gap between "machine" and "computer". In other words, the two words "machine" and "computer" are in two different contexts. We believe that if we restrict the distance between two neighboring elements in a sequential pattern, then this pattern is more meaningful and discriminative. We define a sequential pattern satisfying a gap constraint as follows.

Definition (Gap Constraint and satisfaction): *A gap is a positive integer, 4 > 0. Given a sequence S = {e 0 1, e0 2 , ..., e0 m} and an occurrence o = {i1, . . . , im} of a subsequence X of S, if ik+1 ≤ ik + 4 (∀ik ∈ [1, m − 1]), then we say that o satisfies the 4-gap constraint. If at least one occurrence of X satisfies the 4-gap constraint, we say that X satisfies the 4-gap constraint.*

Note that we consider the subsequences with length 1 (i.e., they contain only one symbol) to satisfy any 4-gap constraint. Hereafter, we call a subsequence X a sequential pattern, meaning that X is a sequential pattern satisfying a 4-gap constraint. Example 3. Let consider an example sequential dataset as shown in Table (). Assume that 4 = 1 and δ = 0.7. The subsequence X = ag is contained in three sequences S1, S2, and S4, and it also satisfies the 1-gap constraint in these three sequences. Thus, its support is sup (X, 4) = 3/4 = 0.75. We say that X = ag is a sequential pattern since sup (X, 4) ≥ δ.

3.4 Phrase Vector representation:

3.4.1 Sequence Embedding:

After associating each sequence with a set of SPs, we follow the Paragraph Vector-Distributed Bag-of-Words (PV-DBOW) model introduced in (Le & Mikolov, 2014) to learn embedding vectors for sequences. Given a target sequence St whose representation needs to be learned, and a set of SPs $F(St) = \{X1, X2, \ldots, Xl\}$ contained in St, our goal is to maximize the log probability of predicting the SPs $X1, X2, \ldots, Xl$ which appear in St:

$$\max \sum_{i=1}^{l} \log Pr(X_i | S_t)$$

Equation 3.2: log probability

Furthermore, $Pr(X_i | S_t)$ is defined by a softmax function:

$$Pr(X_i | S_t) = \frac{\exp(g(X_i).f(S_t))}{\sum_{X_j \in F(D)} \exp(g(X_i).f(S_t))'}$$

Equation 3.3: Softmax function

where $g(Xi) \in R\ d\ and\ f(St) \in R\ d$ are the embedding vectors of the sequential pattern Xi ∈ F(St) and the sequence St respectively, and F(D) is the set of all SPs discovered from the dataset D. Calculating the summation P Xj∈F(D) exp (g (Xj) · f(St)) in Equation 2 is very expensive since the number of SPs in F(D) is often very large. To solve this problem, we approximate it using the negative sampling technique (Jo & Oh, 2011). The idea is that instead of iterating over all SPs in F(D), we randomly select a relatively small number of SPs which are not contained in the target sequence St (these SPs are called negative SPs). We then attempt to distinguish the SPs contained in St from the negative SPs by minimizing the following binary objective function of logistic regression:

$$O_1 = -\left[\log \sigma\left(g(X_i).f(S_t)\right) + \sum_{n=1}^{K} \mathbb{E}_{X^n \sim P(X)} \log \sigma\left(-g(X_i).f(S_t)\right)\right]$$

Equation 3.4: Objective function

where σ(x) = 1 1+e−x is a sigmoid function, P(X) is the set of negative SPs, Xn is a negative sequential pattern draw from P(X) for K times, and g(Xn) ∈ R d is the embedding vector of Xn. We minimize O1 in Equation 3.4 using stochastic gradient descent (SGD) where the gradients are derived as follows:

$$\frac{\partial O_1}{\partial g(X^n)} = -\sigma\left(g(X^n).f(S_t) - \mathbb{I}X_i[X^n]\right).f(S_t)$$

Equation 3.5: derivative of gradient

$$\frac{\partial O_1}{\partial g(X^n)} = -\sum_{n=0}^{K} \sigma\left(g(X^n).f(S_t) - \mathbb{I}X_i[X^n]\right).g(X^n)$$

Equation 3.6: derivate (ii) of gradient

where $\mathbb{I}X_i[X^n]$ is an indicator function to indicate whether Xn is a sequential pattern Xi ∈ F(St) (i.e., the negative sequential pattern appears in the target sequence St) and when n = 0, then Xn = Xi.

3.4.2 Sqn2Vec method:

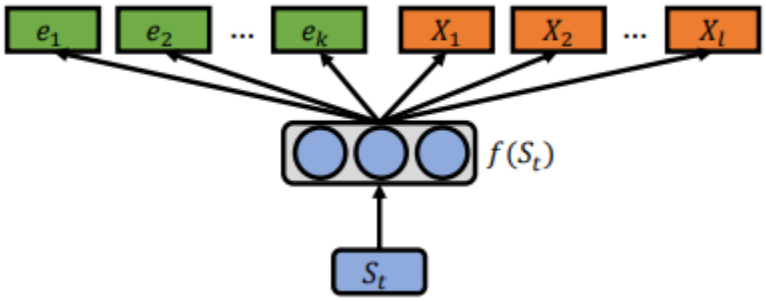

Figure 3.4: Seq2Vec Embedding to obtain vectors

Sqn2Vec-SIM model which uses information of both single symbols and SPs of a sequence simultaneously. The overview of this model is shown in Figure 3. More specifically, given a sequence St, our goal is to minimize the following objective function:

where I(St) is the set of singleton symbols contained in St and F(St) is the set of SPs contained in St.

Equation 5 can be simplified to:

where pi ⊆ St is a symbol or a sequential pattern. Following the same procedure in Section 3.2, we learn the embedding vector f(St) for St, and the embedding vectors of two sequences Si and Sj are close to each other if they contain similar symbols and SPs.

3.5: Latent Dirichlet Allocation

Latent Dirichlet Allocation (Bei et al. 2003) is a Bayesian model which is built on the following assumptions:

- *Word:* the basic unit of discrete data
- *Document:* a collection of words
- *Corpus:* collection of documents
- *Topic (hidden):* a distribution over words & the k number of topics (where k is known)

1. Choose a topic mixture for the document (over a fixed set of K topics).
2. Identify each word in the document by:
    - First picking a topic.
    - Then using the topic to identify the word itself.
    - LDA then seeks to backtrack from the documents to discover a set of themes that are likely to have created the collection, assuming this generative model for a collection of documents.

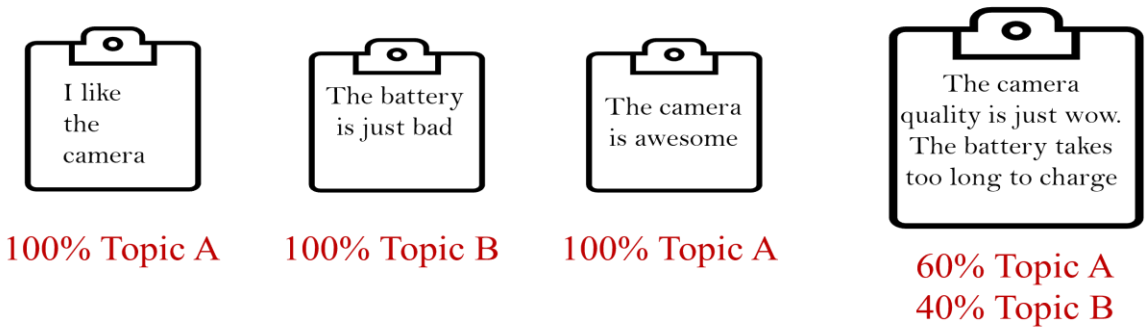

Figure 3.5: Topic modeling using LDA

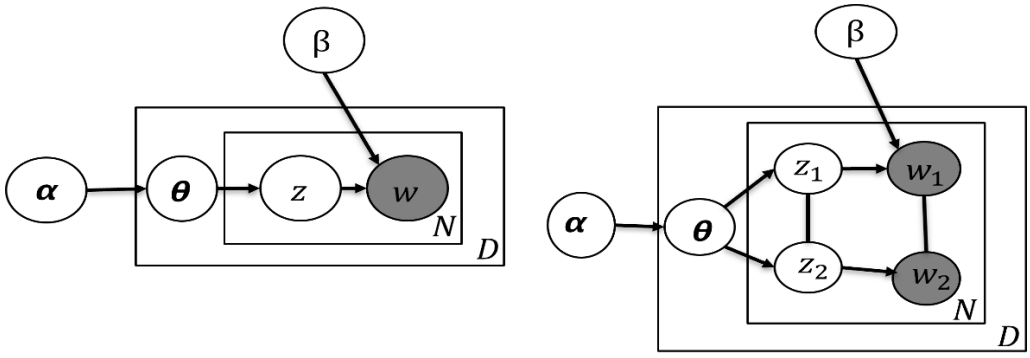

Figure 3.6: (left)- LDA model (right) – working of proposed LDA model

The outer plates in the both the figures represents documents. The major difference lies in the inner plate. Figure 3.6 (left) represents the repeated choice of topics and words within a document. Figure 3.6 (Right) represents the repeated choice of topics and "phrases" within a document.

3.6 A walkthrough Example with comparison from the previous system

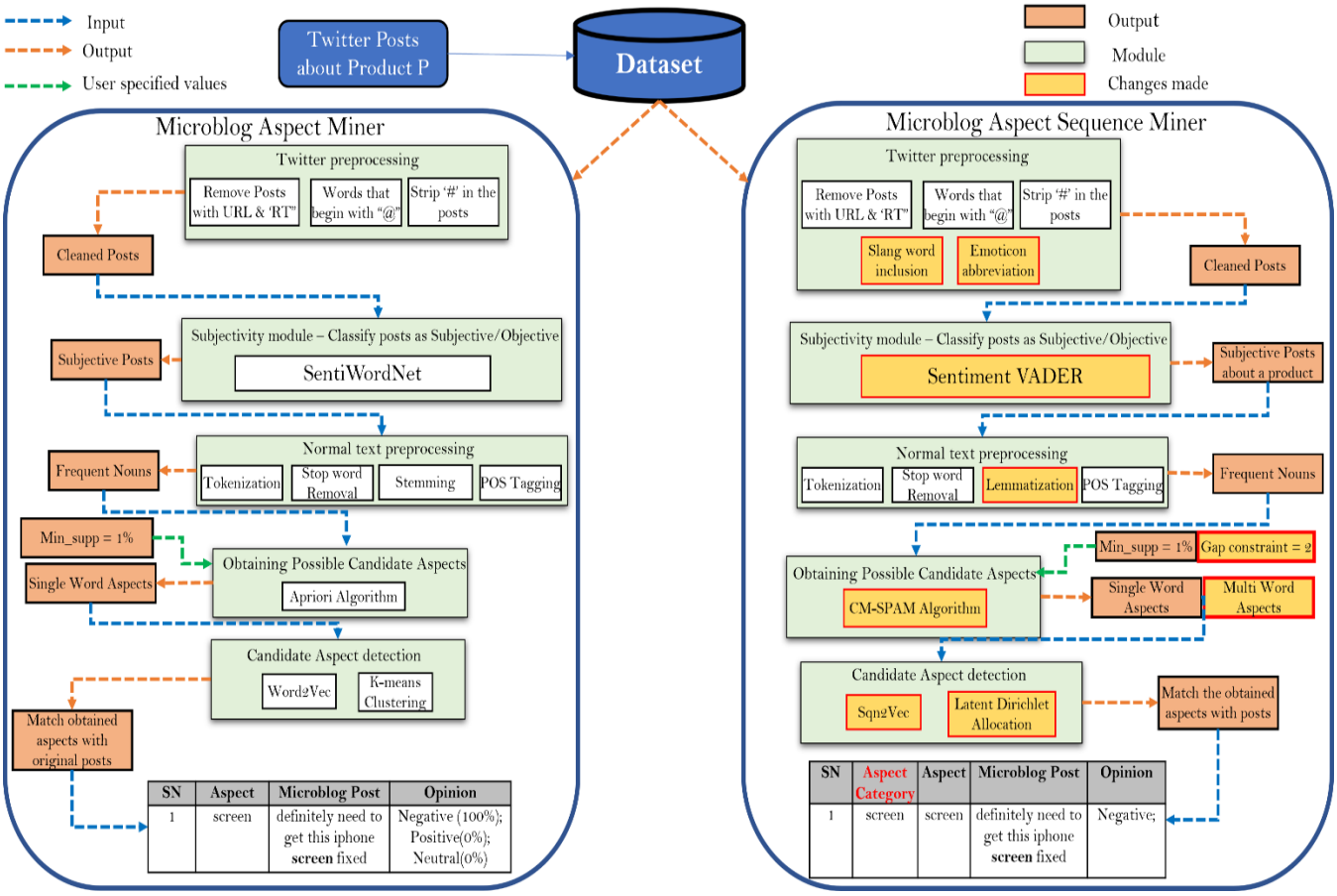

Figure 3.7: Comparison between MAM and MASM

Steps of MAM system

Step 1: Collection of Twitter Data is done through the standard Twitter dataset published

SN	Microblog Posts (Tweets)
1)	RT @xyzzy Definitely have to get this iPhone screen fixed!!
2)	@_kaliblaze iPhone 6 are a pain for phone cases ðŸ˜, I mean why make a phone so thin & not bring out
3)	does anyone have an extra iPhone charger ðŸ˜©
4)	RT @ElaineBaldwin86: lol at my iphone cutting off and not cutting back on @ 89% battery

Step 2: Preprocessing of Twitter - Removes foreign characters, URLs, RT (Retweet).

SN	Pre-processed posts
1)	definitely have to get this iphone screen fixed
2)	iphone are a pain for phone cases i mean why make a phone so thin not bring out
3)	does anyone have an extra iphone charger
4)	lol at my iphone cutting off and not cutting back on 89% battery

Steps of the Proposed MASM System

Step 1: Data collection is done through Twitter API.

SN	Microblog Posts (Tweets)
1)	RT @xyzzy Definitely have to get this iPhone screen fixed!!
2)	@_kaliblaze iPhone 6 are a pain for phone cases ðŸ˜, I mean why make a phone so thin & not bring out
3)	does anyone have an extra iPhone charger ðŸ˜©
4)	RT @ElaineBaldwin86: lol at my iphone cutting off and not cutting back on @ 89% battery

Step 2: Preprocessing of Twitter - Removes foreign characters, URLs, RT (Retweet), emoticons, slang -

SN	Pre-processed posts
1)	definitely have to get this iphone screen fixed
2)	iphone are a pain for phone cases i mean why make a phone so thin not bring out
3)	does anyone have an extra iphone charger
4)	lol at my iphone cutting off and not cutting back on 89% battery

Step 3: Obtain the subjective posts using SentiWordNet (Esuli & Sebastiani, 2014)

SN	Subjective Posts
1)	definitely have to get this iphone screen fixed
2)	iphone are a pain for phone cases i mean why make a phone so thin not bring out
3)	lol at my iphone cutting off and not cutting back on 89% battery

Step 4: Tokenization and stopword removal

SN	Pre-processed posts
1)	‘definitely', ‘have’, 'get', 'iphone', 'screen', 'fixed'
2)	'iphone', '6', 'pain', 'phone', 'cases', 'i', 'mean', 'make', 'phone', 'thin', 'bring'
3)	'lol', 'iphone', 'cutting', 'cutting', 'back', '89%', 'battery', ‘life’

Step 5: Single word aspect Extraction (POS tagging) – Apriori [satisfies frequency requirement]

‘iphone’: 3, ‘phone’: 2, ‘get’: 1 ‘screen’: 1 , ‘battery’: 1, ‘life’: 1,‘cutting’: 2, ‘pain’: 1 , ‘cases’: 1, ‘back’: 1,

Step 3: Obtain the Subjective posts using Sentiment VADER

SN	Subjective Posts
1)	definitely have to get this iphone screen fixed
2)	iphone are a pain for phone cases i mean why make a phone so thin not bring out
3)	lol at my iphone cutting off and not cutting back on 89% battery

Step 4: Tokenization and Stopword Removal

SN	Pre-processed posts
1)	‘definitely', ‘have’, 'get', 'iphone', 'screen', 'fixed'
2)	'iphone', '6', 'pain', 'phone', 'cases', 'i', 'mean', 'make', 'phone', 'thin', 'bring'
3)	'laughing', ‘out’, ‘loud’, 'iphone', 'cutting', 'cutting', 'back', '89%', 'battery'

Step 5: single word Aspect extraction (Pos tagging) – CM SPAM (Fournier-Viger et al., 2014). [satisfies frequency requirement]

[looses the word life].

[phone cases] is a potential aspect.

[battery life] is a potential aspect.

Step 6: Similarity Score using Cosine (Aspect Pruning Method 1) [Formula]

SN	Frequent nouns	Similarity with product
1	iphone	1.000
2	phone	0.7158
3	screen	0.5685
4	get	0.4290
5	pain	0.2353
6	back	0.4164
7	cases	0.4525
8	cutting	0.2401

Step 7: Clustering of pruned frequent nouns.

Cluster 1 = {*get, back*}

Cluster 2 = {*cases, phone, iphone, screen*}

Step 8: Aspect-Product similarity Threshold (Aspect Pruning Method 2)

iphone = {*screen, battery, cases*}

'iphone': 3, 'phone': 2, 'get': 1 'screen': 1, 'battery': 1, 'cutting': 2 'pain': 1, 'cases': 1, 'back': 1,

Step 6: Multi word aspect Extraction – CM SPAM algorithm. (Gap constraint = 2). Represented by $a = a_1 \times a_2 \times a_3 \ldots a_n$.

SN	Frequent noun phrases	Support Count
2	phone cases	1
3	battery life	1

Answer:

i) Finds meaningful aspects compared to MAM and doesn't loose the word 'life'

ii) Preserving the order and meaning of the phrases.

iii) Disadvantage of HCTS or TAC – ['phone cases' or 'cases phone'], ['life battery' or 'battery life'] – Which to choose?

Step 7: Significance Score – Statistical Significance.

Input: Dataset D + Support Count

Step 8: Phrase LDA (Aspect Ranking)

(Question: What is the most common aspect people are talking about?)

Step 9: Aspect based opinion summary

SN	Aspect	Microblog Post
1	screen	definitely have to get this iphone **screen** fixed
2	cases	iphone are a pain for phone **cases** i mean why make a phone so thin not bring out
3	battery	lmao at my iphone cutting off and not cutting back on 89% **battery**

Aspects and the Posts that they occur in.

We obtain a summary of each aspect's opinions from the AOM module, which is the system's ultimate output. The summary of this case is given below:

SN	Aspect of Iphone	Opinion
1	screen	**Negative (100%).** Positive (0%). Neutral (0%)
2	cases	**Negative (100%).** Positive (0%); Neutral (0%)
3	battery	**Negative (100%).** Positive (0%). Neutral (0%)

Final Output of the System

Is it phone cases? Screen? Battery life? – Missing by MAM

SN	Dominant_topic	Topic_name
1	Topic 1	Phone cases
2	Topic 2	Screen
3	Topic 3	Battery life

Step 10: Aspect opinion summary

SN	Aspect of Iphone	Opinion
1	Phone cases	negative: 4 positive: 1
2	Screen	negative: 4
3	Battery life	positive: 3

CHAPTER 4 : COMPARATIVE AND PERFORMANCE ANALYSIS

In this section, we present various experiments to evaluate the efficiency and effectiveness of the proposed approach.

4.1 Dataset Selection

In this section, we present two different datasets widely used in the Twitter sentiment analysis literature. We chose datasets that are

(i) publicly available to the research community,
(ii) carefully annotated, giving a credible set of judgments over the tweets, and
(iii) utilized to test multiple sentiment analysis algorithms.

Tweets in these datasets have been annotated with different sentiment labels, including Negative, Neutral, Positive, Mixed, Other, and Irrelevant. Table 4.1 displays the distribution of tweets in the eight selected datasets according to these sentiment labels.

Dataset	No. of Tweets	#Negative	#Neutral	#Positive	#Irrelevant
Sanders Twitter Corpus	5,113	572	2,333	519	1,689

Table 4.1: Total number of tweets and the tweet sentiment distribution in all datasets

4.1.1 Sanders Twitter Dataset

The Sanders dataset consists of 5,113 tweets on four different topics (Apple, Google, Microsoft, Twitter). One annotator manually labeled each tweet as either *positive, negative, neutral, or irrelevant* concerning the topic. The annotation process resulted in 654 negative, 2,503 neutral, 570 positive, and 1,786 unrelated tweets. The Sanders dataset is available at https://github.com/zfz/twitter_corpus

4.1.2 Twitter-API crawler

To collect Twitter data, researchers typically use the freely available API endpoints for public data. There are two different APIs to collect Twitter data.

(i) The Representational State Transfer (REST) API provides information about individual user accounts or popular topics and allows for sending or liking Tweets and following accounts.

(ii) The Streaming APIs are used for real-time collection of Tweets and come in two flavors:

a. First, the Filter API extracts Tweets based upon a user's query containing keywords, user accounts, or geographic areas.

b. The Filter API is used for studying Twitter content found on a predefined set of topics, user accounts, or locations.

For our research, we have used the (ii)(b) ***Streaming Filter API used on a predefined set of topic and user accounts.***

4.1.2.1 Data Acquisition

In this thesis, we used 100,000 tweets from 4 products and brands from different as our text corpus. The products are Apple, Microsoft, Google, and Twitter. Apple and Microsoft were chosen because they are among the most talked-about products on Twitter, and Starbucks and Sony are easily recognizable brands. We obtained English tweets from Twitter throughout the month. (June 2021 – July 2021).

4.2 Experiment Setup

- Java Programming Language (Eclipse):
  - Preprocessing using Regular Expression.
  - CM-SPAM algorithm.
- Python Programming Language (Google Colab):
  - Twitter API (Tweepy) for crawling the data from Twitter.
  - NLTK for tokenization, stopword removal, POS tagging

- Spacy for Sentiment VADER.
- Latent Dirichlet Allocation.
- Graphs and Visualization.

4.3 Exploratory Data analysis of Sanders Corpus Twitter Dataset

4.3.1 Identifying the top words in the dataset

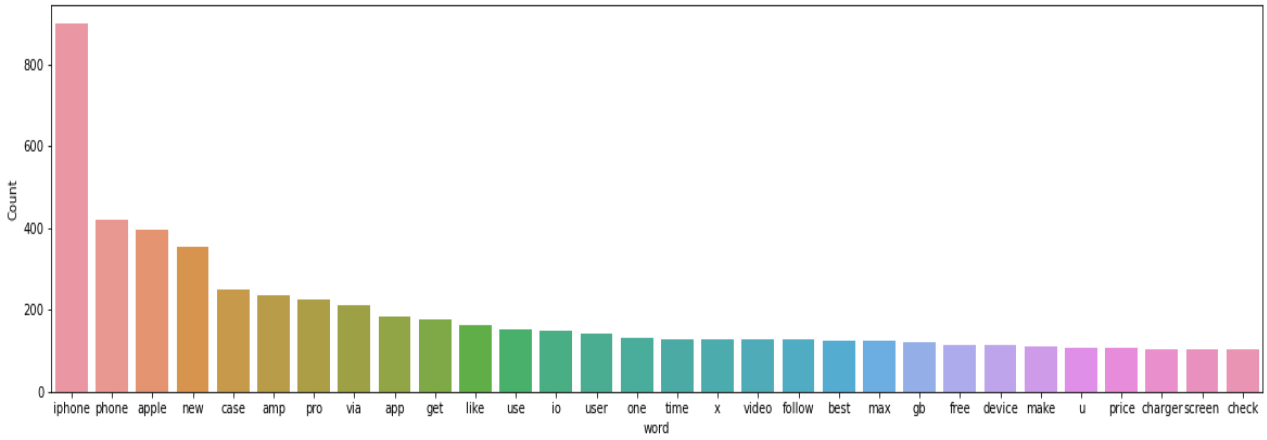

Figure 4.1: Top words after preprocessing

Figure 4.2: Word cloud

In this section, we present a comparison of the described datasets according to three different dimensions: the vocabulary size, the total number of tweets, and the data sparsity.

4.3.2 Vocabulary Size

The vocabulary size is commonly determined by the number of unique word unigrams that the dataset contains. To calculate the number of unigrams, we utilize the TweetNLP tokenizer (Gimpel et al., 2010), designed mainly for Twitter data. Note that all tokens discovered in the tweets were considered, including words, numbers, URLs, emojis, and special characters (e.g., question marks, intensifiers, hashtags, etc.).

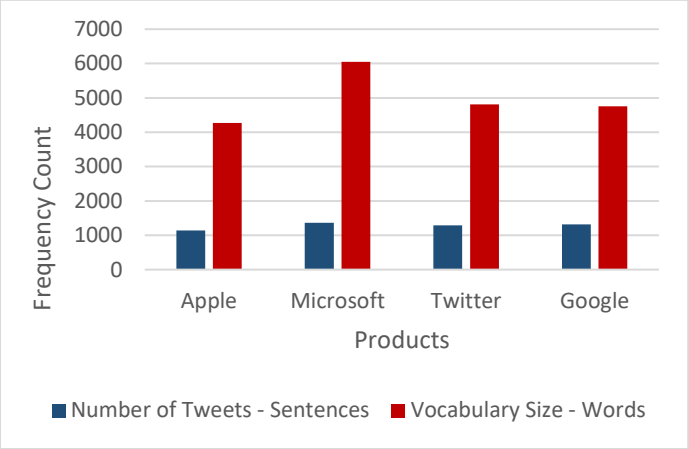

Figure 4.3: Sanders Twitter corpus

4.4 Evaluation Measures

We assess the performance of aspect extraction first, and then the topmost relevant aspects based on those extracted aspects because the goal of the study is to get the most relevant aspects:

4.4.1 Evaluation metrics for Aspect Extraction

We evaluate our proposed model MASM aspect extraction process with MAM, HCTS and TAC with three performance metrics such as: Precision, Recall and F1-measure.

Precision: We calculate the precision to identify the proportion of extracted aspects which are true over the total number of extracted aspects.

$$Precision = \frac{|Extracted\ Aspects \cap True\ Aspects|}{|Extracted\ Aspects|}$$

Equation 4.1: Precision

Recall: We calculate the recall to identify the proportion of true aspects extracted by the system.

$$Recall = \frac{|Extracted\ Aspects \cap True\ Aspects|}{|True\ Aspects|}$$

Equation 4.2: Recall

F1- measure: a measure of a test's accuracy and is calculated using precision and recall as given below:

$$F - Measure = (2 * Precision * Recall) / (Precision + Recall)$$

Equation 4.3: F1-measure

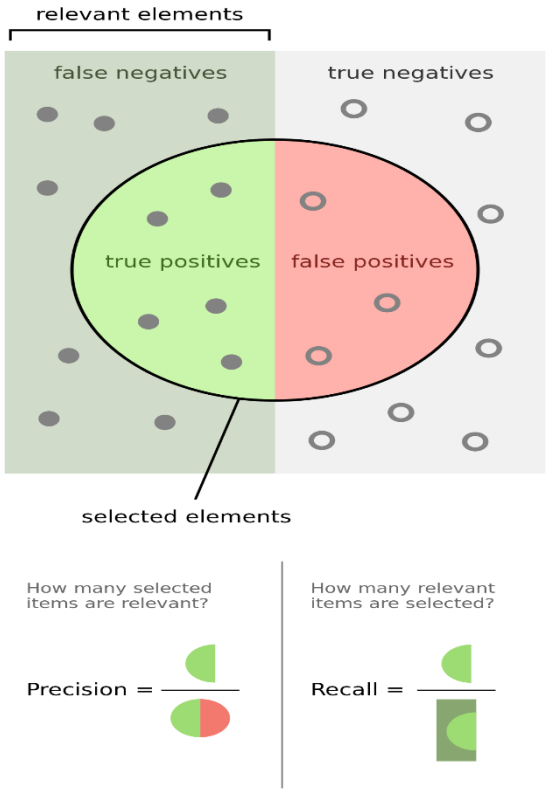

Figure 4.4: Precision and Recall (Wikipedia)

4.4.2 Evaluation of Topic models

In many Natural Language jobs, the scenario is that you have a language L and wish to develop a model M for it. A language L, in this context, refers to a text-generation process. For the sake of clarity, we'll assume we're modelling sentences and the text is made up of sequence words that

conclude in a “word” at the conclusion of the sentence. To generalise to any situation, replace "word" with "token" and "sentence" with "document."

The language L is the likelihood that the next word in a phrase will be w, given a history h of previous words in a sentence:

$$h\ a\ series\ of\ words\ (w_1, w_2, \ldots\ldots, w_{n-1})$$

$$L(w|h) = Prob(next\ word\ is\ w \mid previous\ words\ are\ h)$$

Equation 4.4: Language model representation

With the above model representation perplexity (Blei et al. 2003) can be defined as:

$$Perplexity\ \left(C_{1,N}\right) = \ P\ (C_{1,N})^{-\frac{1}{N}}$$

Equation 4.5: Perplexity calculation

Where N is the number of words, C is the candidate sequence and $P\ (C_{1,N})$ is the probability of the candidate sequence. In simple words, it is the accuracy with which a model predicts the following word (sample).

We want our probability to be high, so that perplexity can be less.

Example: Suppose there are 3 words in a Document D. The probabilities of the 3 characters given by the models are P(battery) = 0.50, P(life) = 0.30, P(power) = 0.20. To determine the perplexity according to the equation:

$$\textbf{Perplexity (“battery life”)} = \frac{1}{\sqrt{P\ (\text{“}battery\ life\text{”})}} = \frac{1}{\sqrt{0.50 * 0.30}} = \textbf{2.63}$$

$$\text{Perplexity (“battery power”)} = \frac{1}{\sqrt{P\ (\text{“}battery\ power\text{”})}} = \frac{1}{\sqrt{0.50 * 0.20}} = 3.22$$

4.5 Results & Discussion

4.5.1 Runtime Comparison

The experiments consisted of running all the algorithms on each dataset while increasing the min_sup threshold until algorithms became very easily executable or a clear winner was observed. For each system, we recorded the execution time, the percentage of candidate pruned by the

proposed algorithms and the total size of CMAPs. The comparison of execution time is shown in Figure 4.5.

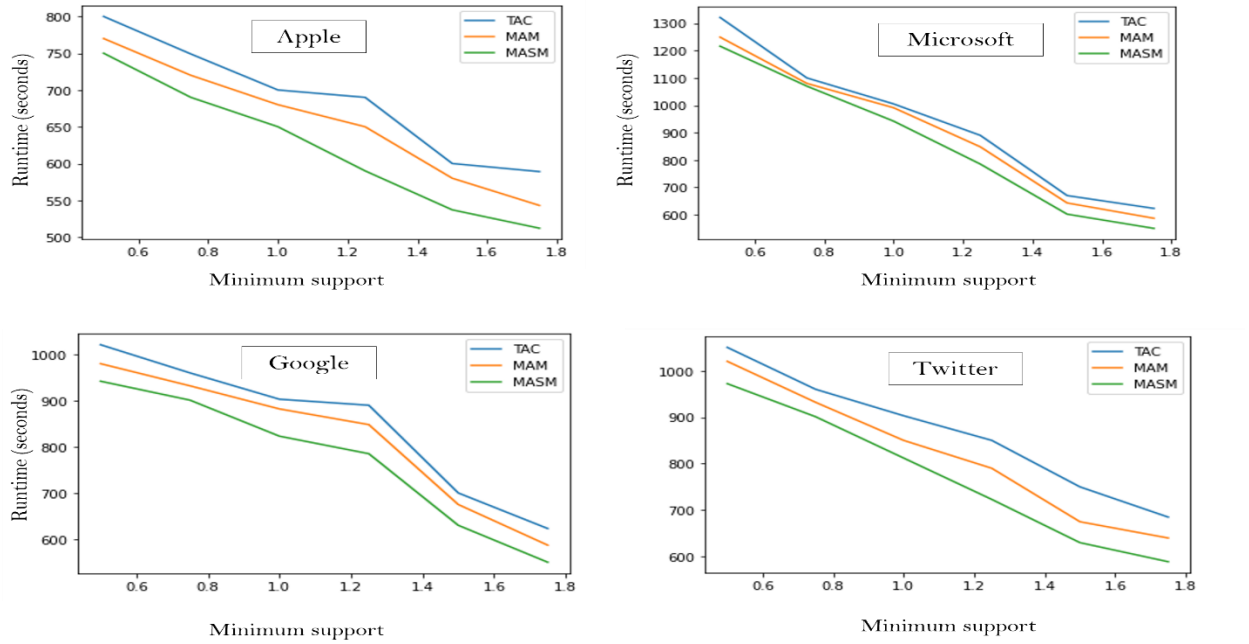

Figure 4.5: Comparison of runtimes between different systems for candidate generation

4.5.2 Results of aspect extraction

Systems	Sanders Twitter Corpus		
	Precision	**Recall**	**F-1 Measure**
TAC	78.5	46.8	58.6
MAM	81.2	61.0	69.66
HCTS	77.9	76.6	76
Proposed MASM	**82.5**	**79.8**	**81.12**

Table 4.2: Evaluation results with different systems

From the following table we can easily see that MAM is performing better than the three other related systems.

(i) The precision of all the four systems is very close because precision as defined in section 4.5.2 is the percentage of extracted aspects as true to the total number of extracted aspects. So, the percentage of all the 4 systems are similar in extracting the relevant aspects. Also,

we can see that MAM and MASM have slightly higher precision the reason being, we remove the neutral statements. If we compare the results of MAM and MASM, MASM has higher

(ii) The recall of MASM compared to all the other systems is relatively higher because we remove the redundant aspects and considers the sequences of aspects which is in the order.

4.5.3 Topic modeling Results

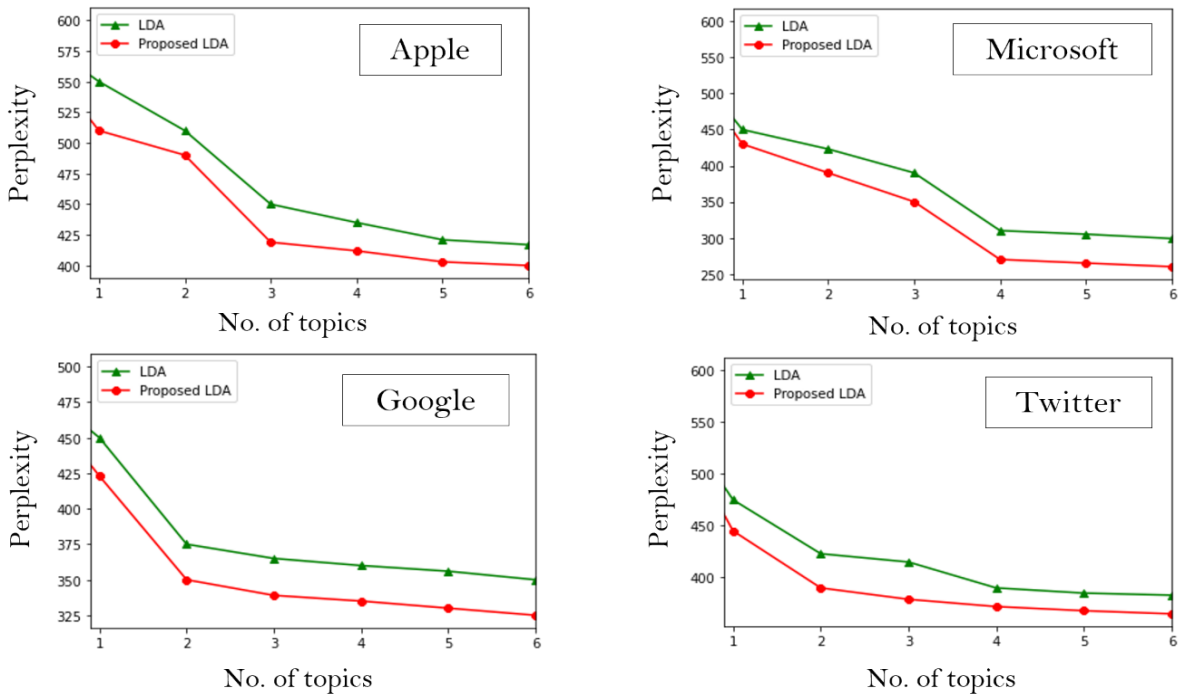

Figure 4.6: Perplexity vs number of topics for different product

Product	Aspect Category	Aspect Terms
Apple	Topic 1	Battery life, battery hour, battery charge, hour life, power supply
	Topic 2	Screen size, screen quality, year warranty, customer service
	Topic 3	Macbook pro, macbook keyboard, macbook air
Twitter	Topic 1	Tweet deck, twitter hiring, tweet space, tweet bot, twitter business
	Topic 2	Easy chirp, media studio
Google	Topic 1	Google ice, ice cream, cream sandwich, galaxy nexus, nexus phone
	Topic 2	Camera app, panorama picture

Table 4.3: multi-word aspects extracted by the proposed LDA method

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

In this thesis, we proposed a hybrid approach Microblog Aspect Sequence Miner (MASM), which generates the multiple word sequences of aspects related to a product. As input, MASM takes in raw unprocessed tweets and first classifies the tweets at the sentence level to determine whether they express any opinion or not. We were able to clean the data required that can be used for sentiment analysis. Then we identify the frequent nouns and noun phrases using a known sequential pattern mining algorithm (CM-SPAM) to determine the possible aspects. This is the stage where we have attempted to improve the existing MAM techniques to generate high-quality phrases using the Sqn2Vec algorithm. Aspect Categorization is an essential task as they represent the opinion targets or what people talk about in opinion. In this study, we have also approached to solve that research problem by modifying the known topic model (LDA), which discovers which categories these aspects belong. Experiments demonstrate that the proposed approach works better in obtaining the relevant aspects of a product with more precision. Getting feedback on these identified elements may also provide business owners insight into what their consumers think of their company. This aids business intelligence and decision-making by answering questions such as, "What portion of my product do consumers like?" and "What part of my rivals' goods do they not like?"

Some of the future work of the system includes:

1) This research only considered the tweets that are expressed in the English language. Identifying aspects in different languages is still a significant limitation of this work. Instead of considering the sequences of nouns as we did, one might think the whole sentence as a sequence and identify the aspects.
2) The datasets that we performed on are purely based on products expressed in Twitter and based on products (Google, Apple, Microsoft, and Twitter). Although we did work on hate crime sentiment analysis, more work is needed on different domains such as political, restaurants, etc. It can also be further enhanced to a different environment such as Amazon reviews and Yelp reviews, where the length of each review is higher compared to microblogs.

REFERENCES

[1] Agrawal, R. & Srikant, R., (1994). Fast algorithms for mining association rules. *Proc. 20th international conference very large databases, VLDB*, 1215, pp. 487-499.

[2] Alsaeedi, A., & Zubair, M. (2019). A Study on Sentiment Analysis Techniques of Twitter Data. *International Journal of Advanced Computer Science and Applications, 10*(2), 361-374.

[3] Al-Shabi, M. A. (2020). Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining. *IJCSNS*, *20*(1), 1.

[4] Amplayo, R. K., Lee, S., & Song, M. (2018). Incorporating product description to sentiment topic models for improved aspect-based sentiment analysis. *Information Sciences, 454-455*, 200–215.

[5] Apté, C., & Weiss, S. (1997). Data mining with decision trees and decision rules. *Future Generation Computer Systems*, *13*(2-3), 197–210.

[6] Baccianella, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

[7] Balazs, J. A., & Velásquez, J. D. (2016). Opinion Mining and Information Fusion: A survey. *Information Fusion, 27*, 95-110.

[8] Bird, S., Klein, E., & Loper, E. (2009). *Nltk: The Natural Language Toolkit*. O'Reilly Media, Inc.

[9] Boyd, D. M., & Ellison, N. B. (2007). Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication, 13*(1), 210-230.

[10] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research, 3*, 993-1022.

[11] Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 440-447.

[12] Carvalho, J., Prado, A., & Plastino, A. (2014). A Statistical and Evolutionary Approach to Sentiment Analysis. *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 110–117.

[13] Chaoji, V., Hoonlor, A., & Szymanski, B. K. (2008). Recursive data mining for role identification. *Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology - CSTST '08*, 218–225.

[14] Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. *Proceedings of the International Conference on Web Search and Web Data Mining - WSDM '08*, 231–240.

[15] Do, H. H., Prasad, P. W. C., Maag, A., & Alsadoon, A. (2019). Deep learning for aspect-based sentiment analysis: a comparative review. *Expert Systems with Applications*, *118*, 272-299.

[16] Dunham, M. H. (2003). *Data mining introductory and advanced topics*. Upper Saddle River, NJ: Prentice Hall/Pearson Education.

[17] Ejieh, C., Ezeife, C. I., & Chaturvedi, R. (2019). Mining product opinions with most frequent clusters of aspect terms. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 546-549.

[18] Evangelista, P. F., Embrechts, M. J., & Szymanski, B. K. (2006). Taming the curse of dimensionality in kernels and novelty detection. *Advances in Soft Computing*, 431-444.

[19] Firmino Alves, A. L., Baptista, C. de, Firmino, A. A., Oliveira, M. G., & Paiva, A. C. (2014). A Comparison of SVM Versus Naive-Bayes Techniques for Sentiment Analysis in Tweets. *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web - WebMedia '14*, 123–130.

[20] Fournier-Viger, P., Gomariz, A., Campos, M., & Thomas, R. (2014). Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information. *Advances in Knowledge Discovery and Data Mining*, 40–52.

[21] Giachanou, A., & Crestani, F. (2016). Like It or Not: A Survey of Twitter Sentiment Analysis Methods. *ACM Computing Surveys*, *49*(2), 1-41.

[22] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., & Smith, N. A. (2010). Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments, 42–47.

[23] Go, A., Bhayani, R. & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *Processing*, 1-6.

[24] Guthrie, D., Allison, B., Liu, W., Guthrie, L., & Wilks, Y. (2006, May). A closer look at skip-gram modelling. In *LREC* (Vol. 6, pp. 1222-1225).

[25] Guyon, I., & Elisseeff, A., "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.

[26] Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques*. Amsterdam: Morgan Kaufmann.

[27] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M.-C. (2000). FreeSpan. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '00*, 355–359.

[28] Hemmatian, F., & Sohrabi, M. K. (2017). A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review, 52*(3), 1495-1545.

[29] Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *In Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 04*, pp. 168–177.

[30] Hu, M., & Liu, B. (2006). Opinion Feature Extraction Using Class Sequential Rules. *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*.

[31] Hutto, C., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and social media* (Vol. 8, No. 1).

[32] Jin, W., Ho, H. H., & Srihari, R. K. (2009). OpinionMiner: a novel machine learning system for web opinion mining and extraction. *In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 09*, 1195–1204.

[33] Jing, H. (2020, November 27). 3 types of sequence prediction problems. Retrieved February 17, 2021, from https://towardsdatascience.com/3-types-of-sequence-prediction-problems-97f22e946318.

[34] Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining - WSDM '11*, 815-824.

[35] Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing*. Harlow: Pearson Prentice Hall.

[36] Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of social media. *Business Horizons, 53*(1), 59-68.

[37] Keshtkar, F., & Inkpen, D. (2012). A hierarchical approach to mood classification in blogs. *Natural Language Engineering*, *18*(1), 61-81.

[38] Lafferty, J. D., McCallum, A. & Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning*, 282-289.

[39] Le, Q. & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. Proceedings of the 31st International Conference on Machine Learning, in PMLR 32(2), 1188-1196.

[40] Lek, H. H., & Poo, D. C. C. (2013). Aspect-Based Twitter Sentiment Classification. *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, 366–373.

[41] Li, F., Han, C., Huang, M., Zhu, X., Xia, Y., Zhang, S., & Yu, H. (2010, August). Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* (pp. 653-661).

[42] Li, S., & Zong, C. (2008). Multi-domain adaptation for sentiment classification: Using multiple classifiers combining methods. *2008 International Conference on Natural Language Processing and Knowledge Engineering*, 1-8.

[43] Liddy, E.D. (2001). Natural Language Processing. *In Encyclopedia of Library and Information Science*, 2nd Ed. NY. Marcel Decker, Inc.

[44] Liu, B. (2010). *Sentiment analysis and subjectivity*. Handbook of Natural Language Processing, second edition.

[45] Liu, B. (2012). *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers.

[46] Liu, B., Hsu, W. & Ma, Y. (1998). Integrating classification and association rule mining. *Proceedings of the 4th international conference on Knowledge Discovery and Data mining (KDD'98)* 80-86.

[47] Liu, B., & Zhang, L. (2012). A Survey of Opinion Mining and Sentiment Analysis. *Mining Text Data*, 415-463.

[48] Mabroukeh, N. R. & Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv*. 43, 1– 41.

[49] Makrehchi, M., & Kamel, M. S. (2008). Automatic Extraction of Domain-Specific Stopwords from Labeled Documents. *Lecture Notes in Computer Science*, 222–233.

[50] Marcheggiani, D., Täckström, O., Esuli, A., & Sebastiani, F. (2014). Hierarchical Multi-label Conditional Random Fields for Aspect-Oriented Opinion Mining. *Lecture Notes in Computer Science Advances in Information Retrieval*, 273-285.

[51] Marrese-Taylor, E., Velasquez, J. D., & Bravo-Marquez, F. (2013a). Opinion Zoom: A Modular Tool to Explore Tourism Opinions on the Web. *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 261–264.

[52] Marrese-Taylor, E., Velásquez, J. D., & Bravo-Marquez, F. (2014). A novel deterministic approach for aspect-based opinion mining in tourism products reviews. *Expert Systems with Applications*, *41*(17), 7764–7775.

[53] Marrese-Taylor, E., Velásquez, J. D., Bravo-Marquez, F., & Matsuo, Y. (2013b). Identifying Customer Preferences about Tourism Products Using an Aspect-based Opinion Mining Approach. *Procedia Computer Science*, *22*, 182–191.

[54] Masseglia, F., Cathala, F., & Poncelet, P. (1998). The PSP approach for mining sequential patterns. *Principles of Data Mining and Knowledge Discovery*, 176–184.

[55] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, *5*(4), 1093-1113.

[56] Meira Jr, W. A. G. N. E. R., & Zaki, M. J. (2014). Data mining and analysis. *Fundamental Concepts and Algorithms*, 1.

[57] Moghaddam, S., & Ester, M. (2010). Opinion digger: A Hybrid Method for Mining Reviews. *In Proceedings of the 19th ACM International Conference on Information and Knowledge Management - CIKM 10*, pp. 1825–1828.

[58] Moghaddam, S., & Ester, M. (2012). Aspect-based opinion mining from product reviews. *In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR 12*, pp. 1184–1184.

[59] Moghaddam, S., & Ester, M. (2013). The FLDA model for aspect-based opinion mining. *In Proceedings of the 22nd International Conference on World Wide Web - WWW 13*, pp. 909–918.

[60] Moraes, R., Valiati, J. F., & Neto, W. P. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications, 40*(2), 621-633.

[61] Nazir, A., Rao, Y., Wu, L., & Sun, L. (2020). Issues and challenges of aspect-based sentiment analysis: a comprehensive survey. *IEEE Transactions on Affective Computing*.

[62] Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., & Schneider, N. (2012). Part-of-speech tagging for Twitter: Word clusters and other advances. *School of Computer Science*.

[63] P. Fournier-Viger, Jerry C. W. Lin, R. U. Kiran, Y. S. Koh & R. Thomas "A Survey of Sequential Pattern Mining," *Data Science and Pattern Recognition*, vol. 1(1), pp. 54-77, 2017.

[64] Pandarachalil, R., Sendhilkumar, S., & Mahalakshmi, G. S. (2015). Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach. *Cognitive Computation,* 7(2), 254-262.

[65] Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04, 271-278.

[66] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02*, 79-86.

[67] Pei, J., Han, J., Mortazavi-Asl, B., & Pinto, H. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proceedings of the International Conference on Data Engineering. 215–224.

[68] Perkins, J. (2010). *Python text processing with Nltk 2.0 cookbook*. Packt Publishing Ltd.

[69] Pham, T. M., Bui, T., Mai, L., & Nguyen, A. (2020). Out of Order: How important is the sequential order of words in a sentence in Natural Language Understanding tasks? *arXiv preprint arXiv:2012.15180*.

[70] Pokou, Y. J., Fournier-Viger, P., & Moghrabi, C. (2016). Authorship attribution using variable length part-of-speech patterns. *Proceedings of the 8th International Conference on Agents and Artificial Intelligence*, 86-91.

[71] Popescu, A. M., & Etzioni, O. (2005). Extracting Product Features and Opinions from Reviews. *In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 339–346.

[72] Porter, M. F., van, R. K. J., & Robertson, S. E. (1980). *New Models in probabilistic information retrieval*. Univ., Computer Laboratory.

[73] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

[74] Rana, T. A., & Cheah, Y.-N. (2016). Exploiting sequential patterns to detect objective aspects from online reviews. *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, 1–5.

[75] Rana, T. A., & Cheah, Y.-N. (2018). Sequential patterns rule-based approach for opinion target extraction from customer reviews. *Journal of Information Science*, *45*(5), 643–655.

[76] Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems*, *89*, 14-46.

[77] *re - Regular expression operations* re - Regular expression operations - Python 3.9.2 documentation. https://docs.python.org/3/library/re.html#re-syntax.

[78] Saif, H., He, Y., & Alani, H. (2012). Semantic Sentiment Analysis of Twitter. *The Semantic Web – ISWC 2012*, 508–524.

[79] Saleh, M. R., Martín-Valdivia, M., Montejo-Ráez, A., & Ureña-López, L. (2011). Experiments with SVM to classify opinions in different domains. *Expert Systems with Applications*, 38(12), 14799-14804.

[80] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613-620.

[81] Sanders, N.J. (2011) Sanders-Twitter Sentiment Corpus. Sanders Analytics LLC.

[82] Sankar, C., Subramanian, S., Pal, C., Chandar, S., & Bengio, Y. (2019). Do neural dialog systems use the conversation history effectively? an empirical study. *arXiv preprint arXiv:1906.01603*.

[83] Scaffidi, C., Bierhoff, K., Chang, E., Felker, M., Ng, H., & Jin, C. (2007). Red Opal: Product Feature Scoring from reviews. *In Proceedings of the 8th ACM Conference on Electronic Commerce - EC 07*, 182–191.

[84] Schouten, K., & Frasincar, F. (2016). Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering, 28*(3), 813-830.

[85] Shu, L., Xu, H., & Liu, B. (2017). Lifelong Learning CRF for Supervised Aspect Extraction. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 148-154.

[86] Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., & Chanona-Hernández, L. (2014). Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, *41*(3), 853-860.

[87] Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Advances in Database Technology — EDBT '96*, 1-17.

[88] Srividya, K., Mirayababu, K., & Sowjanya, A. (2017). Mining Interesting Aspects of a Product using Aspect-based Opinion Mining from Product Reviews (RESEARCH NOTE). *International Journal of Engineering*, 30(11), 1707-1713.

[89] Sutton, C. & McCallum, A. (2011). An Introduction to Conditional Random Fields for Relational Learning. In L. Getoor & B. Taskar (ed.), *Introduction to Statistical Relational Learning*. MIT Press.

[90] Taimoor, M., & Khalid, S. (2018). A Novel Opinion Reason Mining Framework Exploiting Linguistic Associations. *Sixth International Conference on Advances in Computing Communication and Information Technology CCIT 2018*, 6–10.

[91] Turney, P. D. (2001). Mining the web for SYNONYMS: PMI-IR Versus LSA on TOEFL. *Machine Learning: ECML 2001*, 491–502.

[92] Vollmer, C., & Precourt, G. (2008). *Always on advertising, marketing, and media in an era of consumer control.* McGraw-Hill.

[93] *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data.* Data-Centric Systems and Applications. Springer.

[94] Wiebe, J., Bruce, R., & O'Hara, T. P. (1999, June). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics* (pp. 246-253).

[95] Xia, Y., Cambria, E., & Hussain, A. (2015). AspNet: Aspect Extraction by Bootstrapping Generalization and Propagation Using an Aspect Network. *Cognitive Computation, 7*(2), 241-253.

[96] Yang, Z., Wang, Y., & Kitsuregawa, M. (2007). LAPIN: Effective Sequential Pattern Mining Algorithms by Last Position Induction for Dense Databases. *Advances in Databases: Concepts, Systems and Applications*, 1020–1023.

[97] Yin, Y., Song, Y., & Zhang, M. (2017). Document-Level Multi-Aspect Sentiment Classification as Machine Comprehension. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2044-2054.

[98] Younis, E. (2015). Sentiment Analysis and Text Mining for Social Media Microblogs using Open-Source Tools: An Empirical Study. *International Journal of Computer Applications*, 112(5), 44-48.

[99] Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine learning*, *42*(1), 31-60.

[100] Zainuddin, N., Selamat, A., & Ibrahim, R. (2017). Hybrid sentiment classification on twitter aspect-based sentiment analysis. *Applied Intelligence*, 1218–1232.

[101] Zhang, Y., Jin, R., & Zhou, Z.-H. (2010). Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*, *1*(1-4), 43–52.

[102] Zimbra, D., Abbasi, A., Zeng, D., & Chen, H. (2018). The state-of-the-art in Twitter sentiment analysis: A review and benchmark evaluation. *ACM Transactions on Management Information Systems (TMIS)*, *9*(2), 1-29.

VITA AUCTORIS

NAME	Vinay Kiran Manjunath
PLACE OF BIRTH	Bengaluru, Karnataka, India
YEAR OF BIRTH	1995
EDUCATION	BNM Public School, Bengaluru, Karnataka, India (2011) Sri Bhagawan Mahaveer Jain college, Bengaluru, Karnataka, India (2011 - 2013) Jyothy College of Engineering, Bengaluru, Karnataka (2013 – 2017) University of Windsor, Ontario, Canada (September 2019 – September 2021)