CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Rule Based Features Extraction from Citation Context to Find Citation Reasons

by

Qaisar Manzoor

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Computing

Department of Computer Science

2022

I dedicate my dissertation work to my parents, supervisor, and all other teachers. A special feeling of gratitude is for my father, the most unswerving man, I ever know in this world

CERTIFICATE OF APPROVAL

Rule Based Features Extraction from Citation Context to Find Citation Reasons

by

Qaisar Manzoor

(MCS191030)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Ayyaz Hussain	QAU, Islamabad
(b)	Internal Examiner	Dr. Nadeem Anjum	CUST, Islamabad
(c)	Supervisor	Dr. Muhammad Abdul Qadir	CUST, Islamabad

Dr. Muhammad Abdul Qadir
Thesis Supervisor
February, 2022

Dr. Nayyer Masood
Head
Dept. of Computer Science
February, 2022

Dr.Muhammad Abdul Qadir
Dean
Faculty of Computing
February, 2022

Author's Declaration

I, **Qaisar Manzoor** hereby state that my MS thesis titled "**Rule Based Features Extraction from Citation Context to Find Citation Reasons**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Qaisar Manzoor)

Registration No: MCS191030

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Rule Based Features Extraction from Citation Context to Find Citation Reasons**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Qaisar Manzoor)

Registration No: MCS191030

Acknowledgement

Allah (S.W.T), the creator of all universes, deserves all thanks. First and foremost, I thank ALLAH (S.W.T.) for giving me the power, wisdom, and blessings to finish this research. My sincere appreciation to my renowned supervisor, Professor Dr. Muhammad Abdul Qadir. I genuinely thank him for his research support, encouragement, and guidance. He helped me understand the subject. He has taught me, consciously and intuitively, how to do good experiments. I also want to thank the Semantics Research Group for their comments on my research. I am really grateful to my family and friends for their help, encouragement, and support during this Master of Science degree. This is all owing to the affection they offer me every day. I pray ALLAH (S.W.T.) for true prosperity in all fields and knowledge for the benefit of mankind.

(Qaisar Manzoor)

Abstract

Sentiment analysis for different text genres is rapidly growing in the last few years. Same is the case for analyzing citations. In fact, citation sentiment detection has become an attractive task helping researchers to identify shortcomings and detecting problems in a particular approach. Especially including negative citations in the weighting scheme to rank a citation index can be really beneficial. Current approaches on automatic citation analysis assume that the sentiment present in a citation sentence represents the sentiment towards the cited paper. But we believe that citation sentiment is just the beginning towards the more robust analysis of citations in terms of cognitive relationship between the citing and the cited paper. Study reveals that citation sentiment and reason to cite are interconnected, making the citation sentiment analysis even more imperative. In general there are two methodologies to detect sentiments; lexicon-based approach and corpus-based approach. Corpus–based approach required a pre-labeled corpus and machine/deep learning algorithms while lexicon-based approach only use Natural Language Processing techniques to extract sentiment. This research focuses on a hybrid approach using lexicon-based NLP rules to develop feature matrix and applying machine-learning algorithms on the extracted features. For a comparative study, we have applied four machine-learning algorithms (SVM, RF, NB and J48) on an annotated corpus of more than 8,700 citation sentences to classify them in positive, negative, and neutral sentiments. However, the corpus has distributed into multiple classes (positive, negative, and neutral) so instead of accuracy, there is a need to improve the macro-F score. The results show that proposed hybrid technique outperforms existing approaches by achieving macro-F 90% as compared with the existing best score of 74%.

Contents

List of Figures

List of Tables

Abbreviations

ACL	Association for Computational Linguistic
ADJ	Adjective
ADVMOD	Aadverb Modifer
AMOD	Adjectival Modifer
BiLSTM	Bidirectional Long Short-Term Memory
CCRO	Citation's Context and Reasons Ontology
CNN	Convolutional Neural Network
ML	Machine Learning
NB	Naive Bayes
NSUBJ	Nominal Subject
POS	Part of Speech
RF	Random Forest
SCONJ	Subordinating Conjunction
SVM	Support Vector Machine
Tf-IDF	Term Frequency Inverse Document Frequency

Chapter 1

Introduction

1.1 Background

A citation is an expression used to refer to another work in a scientific text, and a reference is a link that indicates the identifier of the cited work. A citation describes the relationship between research papers and all the cited papers make a citation graph of the citing paper with arrow from cited paper to the citing paper. A paper's citations can be counted as citation frequency and if a paper has more citations its citation frequency will be more. The citation frequency is considered an important parameter to classify the paper into important paper. In this case, all citations are counted with the same weight and treated equally [1].

Citation analysis is a study of the citation graph that examines the nature of citations along with the relationship between the citing paper and the cited paper. A citing paper may cite a paper for different reasons. One of the simplest reasons may be the polarity of a citation, for example positive, negative or neutral, name as sentiments of the citation. A considerable amount of research has been done to identify the sentiments from product analyses and newspaper scripts [2]. However, a disproportionately small amount of emphasis has been placed on the extraction of opinions from scientific literature, and more specifically, citations reasons. Polarity exploration of citations graph examines the authors' intention for a particular cited paper. A positive citation may be considered as the strength of the citation that has been referenced, e.g., in case the cited work is being used or extended. A negative citation may articulate a weakness of the cited work. A neutral citation may be a reason for just a piece of background information for a particular idea.

Enrichment of the citation graph with the citation reasons will help the searching of a particular reasoned citation for a researcher. Depending upon the context of citation, the citations is going to convey a useful information to the scholarly community in the form of a meaningful knowledge graph of citations. Therefore, discovery of correct citation reasons is an important task in scientometric or bibliometric analysis. Discovery of citations reasons is being termed as citation classification.

In the last five decades, a number of classification schemes for citations were formulated and developed. Garfield [3] in his research, has listed 15 reasons for citing other people's work and provided a valid signal that the citation and its nature is important. His work paved the way for the analysis of citations. Later, Lipetz [4] established a 4-category scheme defining the relationships between the cited and the citing articles. Afterwards, Athar [5] proposed sentiment analysis of scientific citations. In accordance with the sentimental context of the citations, he divided them into three sentiment classes: positive, negative, and neutral. Furthermore, these sentiment classes were far too simplistic to adequately cover the wide range of reasons for citations classifications in the context of citations. To improve the efficiency of the classification of large amount of data using computer technology, several researchers are working to develop a pattern that could be effortlessly improved by the automatic classifiers.

Ihsan et al. [6] developed a set of eight citation reasons which are formally represented in the form of Ontology named as Citation's Context and Reasons Ontology (CCRO). Machine learning system can be used to identify the reasons for these citations. The eight citation reasons are shown in Table: 1.1. If the CCRO reasons are to be extracted from the citation context, there is a need to identify the context class and then find the reasons within a context class. The accuracy of the next step of finding the reasons depends on finding the context classes. Currently, the state-of-the-art scheme to find citation's polarity [7] gives 76% macro-F

Table 1.1: Citation Reasons

Context Class	Citation Reasons	Collaborative Meaning
Positive	Incorporate	To cite a research as part of a whole
	Extend	To spread from a central research to a wider solution
	Based On	To use a research as foundation or starting point
Negative	Negate	To cause to be ineffective or invalid
	Criticize	To find fault in a research with: points out the faults of
	Contrast	To show differences with opposite nature
Neutral	Compare	To examine in order to show similarities
	Discuss	To consider or examine by argument

using on ACL Anthology dataset [8]. There is a need to improve the precision and recall of finding the context classes (sentiments) in order to have better macro-F. This thesis investigates and demonstrate a system with higher macro-F of citation context sentiment analysis.

1.2 Problem Statement

Macro-F of finding the context class from citation context can be improved by exploiting different language features like nouns, verbs, adverbs, and adjectives. Since of their polarity-bearing nature, adjectives and adverbs are used as modifiers of verbs and nouns respectively, their combined use could contribute significantly to improve the score of finding the sentiment polarity from citation context.

1.3 Research Question

To solve the problem, as indicated in the problem statement, we need to address at least following questions:

1. How important features can be extracted from the citation context by using NLP based rules with the use of nouns, verbs, adverbs, and adjectives (formulation of rules)?
2. Which features give maximum precision and recall (F measure) for a particular classifier?
3. How the usefulness of the selected features can be demonstrated by using state of the art datasets and classifiers?

1.4 Purpose

The goal of this study is to investigate the use of linguistic features, especially adjective and adverbs to improve the precision and recall of the classification of citation context into context classes or sentiment polarity

1.5 Scope

This thesis devises NLP-based rules for the extraction of important features from citation context to increase the macro-F of sentiment analysis.

1.6 Significance

Citation's classifier could be utilized in CCRO system for categorizing the links among the citing and cited papers into citation reasons which can then be used to build a meaningful knowledge graph for the published research papers. The knowledge graph can then be used to answer meaningful queries related with the citation reasons as an application in digital libraries.

1.7 Methodology

Citation polarity analysis tool could be utilized in bibliometric applications for categorizing the links between the citing and cited papers into positive, negative, and neutral classes. As given in the previous chapter, there are three major questions to be answered in developing a higher precision and recall system. The following sections give a brief description used to answer each question.

1.7.1 Selection of Useful Features for a Classifier

A detailed critical survey of the published citations reasons is required in order to answer the question. Several critically important features of citations' content (text) are explored in this research. These features are adjectives, adverbs, and nouns. It is essential to explore the various types of verbs, adjectives, and nouns that can be used to categorize citations into three groups: those that are positive, those that are negative, and those that are neutral.

1.7.2 Formulation of Rules to Extract Useful Features

As we have discussed above, a survey has been conducted and looked for useful features for the classifier. Feature selection rules have been devised to extract important features from citation texts to increase the accuracy of citation polarity analysis. We will prepare the rules by conducting experiments on the annotated data set. We will identify some important parts of speech. These parts of speech include nouns, verbs, adverbs, and adjectives. Thus, these POS helped to figure out the reasoning class of citations.

1.7.3 Demonstration of the Usefulness of Features for Classification

A classifier is required to evaluate appropriate features to classify citation polarity. We will use the same machine learning techniques (as used by the state of art systems) to categorize the citation text into three classes. The machine learning method requires an annotated dataset, and we already have two different annotated datasets. Thus, these datasets will be used by the machine learning classifier to classify the citation texts. The classifier trained on annotated data set when it comes to testing the classification results of the citations extraction system, the k-fold cross-validation technique incorporated.

Chapter 2

Literature Review

The problem is the identification of the citation polarity from citation context. Here, the major schemes to find citation polarity will be reviewed against the problem and the research question raised in the previous chapter.

2.1 Selection of Useful Features

Garfield [3] took the initiative in 1965 to determine the reason for the citation. (1965) when he was trying to automatically index a citation in the text. He mentioned 15 reasons to depict that why the authors make a reference to cite another document. These reasons include paying tribute to pioneers, crediting related work, identifying methods and equipment, providing background knowledge, correcting others' work, criticizing previous work, identifying original publications in which ideas were discussed, identifying the original publication defining an eponymous concept or term, and so on. His focus was not to extract the reasons from the citations; therefore, nothing was discussed about the features for identification of reasons from citation context.

Moravcisk and Murugesan [9] proposed an overlapping four-part categorization of citations in 1975 (Organic Perfunctory, Confirmative Negational, Evolutionary Juxtapositional, and Conceptual Operational). The author gathered a random sample of data from Physical Review, which was published during 1968 to 1972.

There were 30 publications with 702 citations. Social scientist (annotators) discovered that 41% of citations fall into the "perfunctory" class, while 14% fall into the "negative" class. The author didn't use features for identification of reasons from citation context. He didn't perform automatic analysis of citations, he just found reasons with a manual process.

Garzone developed a pragmatic grammar with 195 lexical matching rules and 14 parsing rules [10] to classify citations. He classified citations according to the article's cue words and location. A classifier for automated citations has been developed, which assigns 35 classes to the citations. The classifier needs syntactic information, such as verbs and nouns. They used a trainable rule-based tagger [11]. Garzone compared their classifier results with manually annotated citations. The average percentages of entirely correct, partially correct, and entirely incorrect citation assignments are 78 percent, 11 percent, and 11 percent, respectively, while the average percentages of entirely correct, partially correct, and entirely incorrect citation assignments are 84 percent, 8 percent, and 8 percent. They didn't use any parameter (F1 score or macro-F) in their experiments.

Teufel [12, 13] utilized the parts of speech tags (verbs) and cue phrases (1762 cue phrases) for identification of grammatical subjects and further categorized as different agent forms. They categorized citations into weak, positive and neutral. They compiled 360 conference papers on computational linguistics from the Computation and Language E-Print. They used a total of 116 articles, randomly drawn from the part of corpus, contains 2829 manually tagged citations. The Instance Based Learner (IBk) technique [14], with k=3, was used to get a 0.71 Macro-F, indicating that their system produced 29% of the incorrect results for their own annotated corpus using their approach.

Sugiyama [15] classified citations into citing and non-citing categories using the support vector machines (SVM) classification technique [16] in 2010. He made use of n-grams (unigrams, bigrams), nouns, and previous and next sentence (If the preceding sentence is a citation, the following sentence may refer to the same work and is therefore less likely to contain an additional citation). They used the ACL Anthology Reference Corpus which contains 955755 sentences from 10921 articles with 112,533 positive instances and 843,242 negative instances. The outcome of study revealed that the SVM classifier achieved accuracy of 0.88.

The citations were grouped by Athar [5] into three categories, positive, negative, and neutral. He built a corpus of 8,736 instances. He used features like parts of speech (adjectives), dependency relations (long distance relationships between words), n-grams (unigrams and bigrams), and science specific sentiment lexicon (effective, efficient, popular, successful, and state-of-the-art). The Support Vector Machine (SVM) and the Naive Bayes (NB) [17] were used, and results were reported using SVM compared to NB. The author created his own dataset. The researcher utilized macro-F metrics for the evaluation of the performance of citations context classification. The approach achieved a macro-F of 0.764. This implies that their system produced on average 24% of the incorrect findings for their own annotated corpus.

Tandon and Jain in [18] categorized the citation text into five distinct classes: (applications, limitations, related work, summary, and strengths). They used Random k-Label sets with Naive Bayes algorithm [19] in their experiments. For every class, they made use of features such as verbs, combinations of adjectives, as well as n-grams. Since no annotated dataset exists, they used Microsoft Academic search engine for 30 research publications and annotate 500 citation contexts. The experimental results indicate that the arrangement of adjectives, verbs, and bigrams averaged 68.54 percent of precision. Precision is not the perfect metric for determining a system's overall performance.

Sentences were divided into three classes by Athar and Teufel [20]. These classes are (objective, negative, and positive). The authors held the opinion that the words and sentences surrounding the citation position contained valuable opinions. These opinions could improve the results of detection of the author's purpose in citing works. He built a new citation corpus and used only 20% of Athar's [5] dataset. They annotated 1741 sentences. They used different features like n-grams (unigram, bigram, trigram) and dependency relations (long distance relationships between words) in three classes. The approach achieved 0.73 macro-F. While they have tested their methods by using SVM classifier. This implies that their system produced very low results for their own annotated corpus. Their work becomes a domain dependent because they have focused on computational linguistic papers.

Parthasarathy et al. [21] extracted citing sentences by using a sentence parser from the data base of Google scholars. They categorize sentences into three classes such as negative, positive, & neutral. They used adjectives as a feature which can be either positive or negative. They proposed that if a sentence lacks an adjective, the sentence would either neutral or unknown. They used different Algorithms of machine learning (ML) namely, J48 [22] and NB detect the polarity of citations. They achieved an 84% F1 score of the totally classified positive sentences. They didn't compute the results of two other classes.

Hernandez-Alvarez and Gomez [23] have recently proposed a new annotation methodology to label sentences of citation into six classes such as (useful, contrast, acknowledge, based on, hedges, and weakness). They have employed features like semantic patterns and n-grams (unigrams, bigrams) to find citation reasons. The author gathered 85 articles randomly from ACL Anthology. They have developed their own corpus. SVM was tested on corpus and have achieved 0.87 of F1 score.

Butt et al. [24] classified the sentences into two classes positive and negative. They used features like regular expressions (to form tokens of sentences), sentiment lexicon (contains key words classified as positive or negative) and phrases of words. They have downloaded 150 research papers from Google Scholar and annotate them manually. They classified the citations using the Naive Bayes classifier in their work by selecting a five-sentence frame from the cited text with a F1 Score of 80%. They used generalized lexica to make experimental findings (from Evert [25] with 28,000 positive and 31,000 negative words) that combine the citations from the multiple fields.

Xu et al. [26] concentrated on 285 clinical trial papers in the discussion section and established a rule-based method for citation extraction. In addition, three annotators manually annotated more than 4000 citations. They used features like n-grams (unigrams, bigrams and trigrams), sentiment lexicon (This lexicon has 53 positive and 46 negative phrases for biomedical research papers.) and structure features (sentiment words, negation words, comparative relation etc.) to classify the citation sentences into three classes based on their positivity, negativity, and neutrality. Combining all these features their system achieved 0.71% macro-F with Support Vector Machine (SVM). This shows that their system generated very low results for their own annotated corpus.

Kim and Thoma [27] came up with an automated citation polarity revealing method using machine learning procedures. The authors presented a method for citation text classification based on support vector machines and n-grams (unigrams and bigrams) as an introductory work, and word statistics as a feature vector. The text is classified into two types using the projected citation polarity classification technique: positive and others. They extracted 2,665 sentences from 414 distinct biomedical papers published online and indexed in MEDLINE. They used size of the word dictionary 500 and successfully achieved 0.80 F1 Score that is comparatively less with respect to other systems, and they classified citations in only two classes (positive and others).

Zheng [28] tackled the problem of polarity classification by using information about a reputation of an author. Tf-IDF was one of the features they used (Term Frequency Inverse Document Frequency), author's ID, polarity distribution and p-index. They have used Athar's [5] dataset for their experiments and achieved 0.53 macro-F. They used Support Vector Machine (SVM) for their experiments. However, their research still requires technical skills to use better features to detect the polarity of scientific citation.

Jha and Abu-Jbara et al. [7] used supervised sequence labeling to determine the citation context of a reference and related adjacent sentences, categorizing citations into three categories or classes: Positive, negative and objective. The style of references in the journals is different which can affect feature extraction. So that is why to clean the context of a citation a regular expression was employed. They used SVM, Logistic Regression, and NB classifiers that had the following characteristics: self-citation, adverb, verb, reference count, adjective, dependency relations & negation. They selected thirty papers from ACL Anthology Network (AAN) and annotate them manually and received 3500 citations. The outcome of study revealed that the SVM classifier achieved results with 0.74 macro-F. Ravi

[29] presented a new feature engineering technique for citation polarity analysis and they used different features like n-grams (unigrams, bigrams, and trigrams), dependency relation (obtained from given sentence) with word vector based convolutional neural network (wvCNN) [30][51] model to extract word vectors. They used two datasets. Athar's [5], and second developed by author by collecting articles from Science Direct[1]. Corpus contains 1125 positive, 181 negative, and 7518 neutral citation sentences. They performed experiments on both datasets and achieved 54.5% macro-F on Athar's dataset and 37.12% macro-F on their own dataset.

Ikram [31] proposed an aspect-based citation polarity analysis framework to classify sentences into one of three categories: negative, positive, or neutral. They extracted the citation sentences aspects using linguistics phrase patterns. Additionally, he made use of SentiWordNet [32] (an opinion lexicon derived from the WordNet database) by considering the words that are used in the linguistic expression of the aspect. They used two datasets, 1) Athar's [5] dataset, 2) Xu et al. [26]. They used different features like POS (nouns, plural nouns, proper nouns (single and plural), adjectives, determiners, and gerunds in verbs), considering bigrams, trigrams, and pentagrams based on n-gram features (N-gram after, N-gram before, and N-gram around). They achieved 85% F1 Score with SVM classifier.

Sula and Miller have developed a tool for recognizing the sentences of citation and to identify the sentiment of the research article in humanities domain [33]. They used NB algorithm with n-grams model for citation classification. To extract the sentences of citation, four humanities journals were used. They annotated a few sentences into dual classes, negative and positive then trained NB classifier to categorize the polarity of citation. They processed 5,700 citation contexts and found only 176 examples of positive sentiment and 58 examples of negative sentiment. They didn't used any metric in result compilation. Jochim and Schutze [34] classified citations to determine their polarity. They used features like dependency relations, n-grams (unigrams, bigrams and trigrams), and cue words and phrases.

[1] www.sciencedirect.com

They used a dataset of overall 2008 citations from the ACL anthology, 1836 were positive and 172 were negative. This procedure scored 68.2% F1.

Yousif et al. [35] proposed a model to address citation sentiment by encoding citation sentiment information in a word embedding representation, which is subsequently used by the neural network of representation learning network (Multitask-RCNN). N-gram features are extracted, and long-term dependencies of the input citation sentence were captured by using a combination of two neural networks: convolutional neural network (CNN) and Bidirectional Long short-term memory (BiLSTM). They used the ACL Anthology Network (AAN), which contains 3568 citations. The F1 of 87.00% was obtained.

It was proposed by Chen et al [36] that the fine-tuning approach should be implemented using the Universal language model fine-tuning for text classification (ULMFiT), BERT, and XLNe to classify sentences into three different classes were used. With 400 embedding layers and 1150 hidden activations per layer, the AWD-LSTM model was used for ULMFiT. There were 12 layers, 768 hiddens and 12 self-attention heads in the BERT-base model while for XLNet, they used a model with 12 layers, 768 hiddens and 12 heads. Two datasets were analyzed. A total of 1768 citation sentiment examples can be found in DFKI [37] while UMICH [7] contains 3568 examples. On ULMFit, BERT, and XLNet, their proposed models scored 88.0% F1, 90.0% F1, and 91.00% F1, respectively.

For the categorization of citations, Mercier et al. [38] presented a neural network design (ImpactCite). With the help of a vast amount of data, XLNet [39] has been pre-trained on an auto-regressive language model that includes bi-directional attention. The ability to recognize relationships inside phrases that may be traced from right to left is a huge benefit of this system. Different XLNet implementations have different number and unit of layers and units. They used two XLNet-Large models in our research where they opted for the big edition of XLNet. 24 layers, 1024 hidden units, and 16 heads make up XLNet-Large. For their experiments, they used Athar's [5] dataset. They achieved 77.00% macro-F.

2.2 Critical Analysis of Literature Review

Different polarity bearing features are incorporated in literature to extract the polarity of citations. Various authors used various feature extraction techniques such as dependency relations, negations, and n-grams (unigram, bigram, and trigram). Different parts of speech (POS) are also adopted to extract the polarity of citations like verbs, adjectives, adverbs etc. Parts of speech are used in literature to extract citation context, but all of the techniques mentioned above did not give satisfactory results. Athar [5] used various feature extraction techniques such as word level features, polarity bearing phrases, negations, and dependency structure. Their implementation only achieved 76% macro-F. This implies that their system produced 24% of the incorrect findings for their own annotated corpus.

In 2016, Athar [20] performed experiments on a new annotated corpus, and this time he adopted different context windows, but he achieved only 73% of macro-F. Ikram [31] used SentiWordNet by considering the words that are used in the linguistic expression of the aspect. They also used different features like POS (nouns, plural nouns, proper nouns (single and plural), adjectives, determiners, and gerunds in verbs), considering bigrams, trigrams, and pentagrams based on n-gram features (N-gram after, N-gram before, and N-gram around). Their linguistics phrase patterns were not good enough to achieve good results, and their implementation only achieved 85% F1 Score.

Table 2.1: Critical Analysis of Existing Approaches

Scheme	**Results**	**Strengths**	**Weaknesses**
Athar [5] 2011	Macro-F: 0.764	Best results with combining ngrams and dependency relations.	Time consuming for feature extraction, did not handle implicit citation, citation annotation (manually), did not compare results with other approaches. Low results.
Athar [20] 2012	Macro-F: 0.731	Using citation context length (explicit and implicit),Improved results with different context windows, Best results with combining ngrams and dependency relations.	Time consuming for feature extraction, citation annotation (manually), only Focused on computational linguistic papers.
Xu et al. [26] 2015	Macro-F: 0.719	Combination of n-grams (uni-gram, bi-gram and tri-gram) and also incorporate sentiment lexicons (bio-medical specific lexicons) features to achieve better/good results.	Annotation process is conducted manually. Citation analysis only for biomedical publications. Their proposed approach did not perform well on other citation (Scientific).

Table 2.1 - Continued from Previous Page

Scheme	**Results**	**Strengths**	**Weaknesses**
Ma et al. [28] 2016	Macro-F: 0.645	To improve H-index method by including negative polarity in the calculation process.	Their research still requires technical skills to use better features to detect the polarity of scientific citation. Citation annotation (manually).Did not compare results with other approaches.
Jha-et al. [7] 2017	Macro-F: 0.74	Best results with dependency relations. Reference count and closest verb, adjective and adverb to the target reference .	Only Focused on computational linguistic papers. Citation annotation (manually). 26% of the incorrect findings in their annotated corpus
Ravi et al. [29] 2018	Macro-F: 0.54	Proposed a novel feature engineering method for citation sentiment analysis to be employed with deep learning	Deep learning models are data hungry models so a large corpus is required for citation polarity analysis in deep learning systems. For analysis, a larger citation window must be examined. Did not handle implicit

Table 2.1 - Continued from Previous Page

Scheme	**Results**	**Strengths**	**Weaknesses**
Ikram et al. [31] 2019	F1 Score: 0.83	They used different features like POS (nouns, plural nouns , proper nouns (single and plural) , adjectives), considering bigrams, trigrams, and pentagrams based on n-gram features.	Used SentiWordNet (an opinion lexicon derived from the WordNet database) didn't achieve good results. Did not compare results with other approaches.
Yousif et al. [35] 2019	F1 Score: 0.87	Proposed Multitask-RCNN model which encodes citation sentiment in word embedding. Proposed model composed of two neural network which are CNN and BiLSTM for extracting n-gram features.	A large corpus is required for citation polarity analysis in deep learning systems. Only focused on n-grams features. For analysis, a larger citation window must be examined.
Chen et al. 36 2020	F1 Score: 0.81 F1 on ULMFit, 0.90 F1 on BERT and 0.91 F1 on XLNet	Proposed fine-tuning approach is based on ULMFiT, BERT and XLNe to classify sentences into three different classes. Used BERT-base model with a hidden size of 768, 12 layers and 12 self-attention heads.	A large corpus is required for citation polarity analysis in deep learning systems. Annotation process manual. Low results.

Table 2.1 - Continued from Previous Page

Scheme	**Results**	**Strengths**	**Weaknesses**
Mercier et al. [38] 2021	Macro-F: 0.77	Used two XLNet-Large models. Used the large version of XLNet. XLNet-Large consists of 24-layers, 1024 hidden units, and 16 heads.	Low results. A large corpus is required for citation polarity analysis in deep learning systems.

Xu et al. [26] used features like n-grams (unigrams, bigrams, and trigrams), sentiment lexicon, and structure features to classify the citation sentences, but their system achieved 0.71% of macro-F. This shows that their system generated 29% of the wrong findings for their own annotated corpus, which is a significant amount of error. Ma et al. [28] used Tf-IDF (Term Frequency Inverse Document Frequency) as one of the features, author's ID, and polarity distribution to classify citations, but their system achieved a 0.53 macro-F score. However, their research still requires technical skills to use better features to detect the polarity of scientific citation.

Jha et al. [7] used features like adverb, verb, reference count, adjective, dependency relations & negation but achieved results with 0.74 macro-F. This indicates that their system generated 26% of the incorrect findings in their annotated corpus. Ravi [29] proposed a novel feature engineering method for citation sentiment analysis to be employed with deep learning and achieved 0.54 macro-F. A large corpus is required for citation polarity analysis in deep learning systems. For analysis, a larger citation window must be examined.

Critical analysis of the literature revealed that the F1 score for the sentiment analysis of citations' context is not very good. There is a potential to use NLP techniques to extract relevant features and then use those features for the classification. This thesis is going to analyze the process of citation sentiment classification based upon the NLP techniques.

Chapter 3

Research Methodology

It is clear from the literature that different approaches used different feature extraction techniques, e.g., ngram, dependency relations, polarity lexicon, and different POS (nouns, plural nouns, proper nouns) as features to find the citation reasons. Some systems used different deep learning models to extract features from the citation, but all of them have failed to obtain satisfiable results. This research focuses on a hybrid approach using lexicon-based NLP rules to develop feature matrix and applying machine-learning algorithms on the extracted features.

The whole process has been completed in three steps. First of all, we selected Athar's and Clinical Trial's data set. In the second step, feature selection rules were devised for extraction of important features from the citation texts. Finally, in the third step evaluation was performed by using different machine learning algorithms. In this way, the entire process was completed, and a worldly-wise strategy has been made to analyze the impact of different parts of speech. A detailed architecture diagram depicts all of the processing steps, as shown in figure 3.1.

3.1 Dataset Collection

Two distinct datasets from the domain of computer science and bioinformatics are used for the studies. We used a particular type of the ACL Anthology Network (AAN) data set designed and annotated by Athar's [5], which contains 8,736 citation sentences categorized as Citing Paper ID, Cited Paper ID, Citation Text, and three sentiment classes which are positive, negative, and neutral. There are 829 positive citations, 280 negative and 7,627 objective citations. The second dataset is derived from clinical trial publications and contains citation lines collected from 285 randomly chosen publications [26]. The collection contains 4182 citation sentences from clinical trial papers calling for biomedical study replication. There are 3172 neutral examples in total and 702 positive and 308 negative citation sentences.

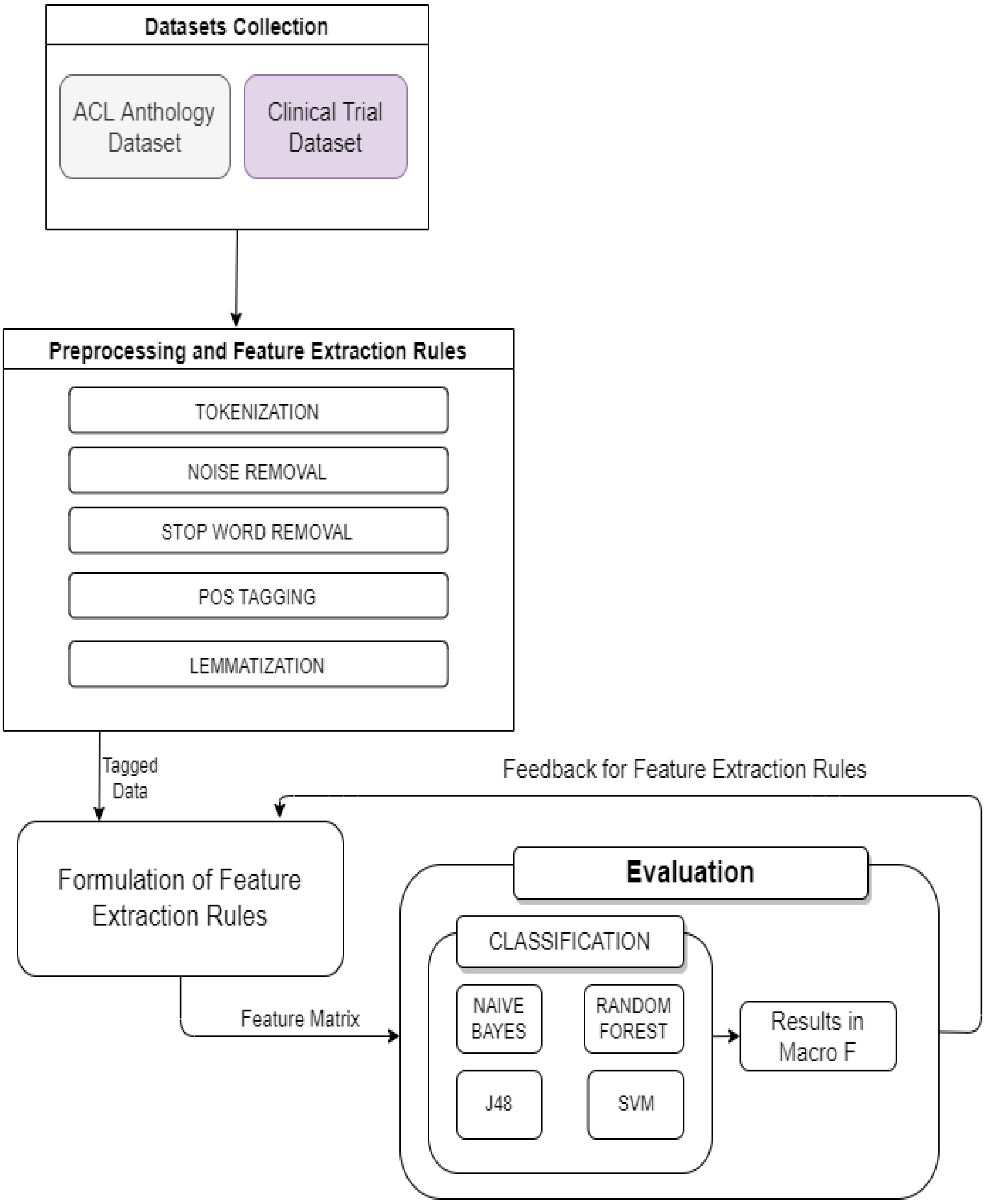

Figure 3.1: Research Methodology

3.2 Pre-Processing

Pre-processing has vital importance since it removes unnecessary and noisy data from the data set. Several different steps have been implemented for the pre-processing process including tokenization, noise removal, stop word's removal, lemmatization and POS tagging. All these steps are discussed one by one in the following sections:

3.2.1 Tokenization

In this step, the citation texts can be divided into meaningful pieces. These pieces are known as tokens. For example, we can split a chunk of text into words or split it into sentences. We split the citation texts into words. Spacy [40] for Tokenization has been used as the best-known and most widely used Natural Language Processing library.

3.2.2 Noise Removal

It is important to remove noise from data because it can adversely affect accuracy. The data sets generally contain noise such as unnecessary punctuation and null values. Spacy provides different functions to remove noise, e.g., length(), size(), remove(), etc. Partial or not-complete sentences are removed.

3.2.3 Stop Words Removal

Stop words are the most frequent words in linguistic theory, such as on, of, an etc. Stop words have no significant meaning, so they can be removed from the citation texts for correct measurement. Spacy library compares the tokenized list to its list of stop words, then stop word removed from the corpus. This step helps reduce the dimension of a features space.

3.2.4 Lemmatization

Lemmatization is a way to reduce words to their roots or basic words. The benefit of lemmatization is that it decreases the size of vocabulary. For example, all the terms like program, programs, programmer, programing, and programmers are lemmatized into their root word program. We have done this using Spacy library that transforms each word of citation texts to its root words. For all citation texts the lemmatization algorithm is applied.

3.2.5 POS Tagging

Parts of Speech (POS) tagging is used to eliminate ambiguity by tagging different words. We can find the lingual features very easily with POS tags. These features include adjective, verb, adverb, noun and their distinctive types. Lingual feature (polarity words) are taken out from POS tagged words, that are used in Machine Learning algorithms. Researchers [5–8] have been using Part-Of-Speech tags to discover the features which catch the sentiment. We extracted important linguistic features from POS tagged words, which are used for classification of citation texts.

3.3 Feature Extraction Techniques

In the classification of citation texts, features extraction is an important technique. Polarity of citations can be extracted by using these techniques. In the previous chapter, we have discussed several types of features for citation analysis. Here, we describe the most common features used for the classification of citation texts in our system.

3.3.1 N-grams

An n-gram is a group of related terms in a text. The character '**n**' denotes the length of the sequence. When n equals one, the series is called a unigram; when **n** equals two or three, it is called a bigram or trigram, respectively. The n-grams are described for sample sentences in the following example.
Example: Ali is a good student
unigrams: Ali, is, a, good, student

bigrams: Ali is, is a, a good, good student

trigrams: Ali is a, is a good, a good student

4-grams: Ali is a good, is a good student

Classification tasks for citations, length 1 and length 2, N-grams performed well [5–9, 41]. Bigrams with adjectives and adverbs are considered more sentimental in addition to looking specifically for the scope of negation words [42]. N-gram lengths from 1 to 2 have been used in our experiments with window size seven.

3.3.2 Bi-Tagged

Bi-tagged type features are obtained by POS tagging. The information based on POS is utilized for extracting sentiment-rich features, although adjectives and adverbs have been investigated in literature, the nature of these are subjective. Turney suggested a methodology [2] for extracting bi-word sentiment-rich features in which each member is either an adjective or an adverb, for example, adjective-noun, adverb-adjective, noun-adjective, adverb-verb etc. It is observed that the verb (verb-noun, verb-adjective, adjective, and adverb-verb) can provide reasoning information useful for citation reasons classification, too.

3.3.3 Dependency Features

Dependency features describe the grammatical relation between the words. Each feature in the dependency structure signifies a binary relationship among *Head Word & Dependent Word.* Generally, dependencies described as triples form relation (*Head Word, Dependent Word*). As it is illustrated in the following sentence and also presented in Figure 3.2.

Sample Sentence: Our system significantly outperforms competing approaches.

The above sentence contains four tokens corresponding to the resulting triplets.

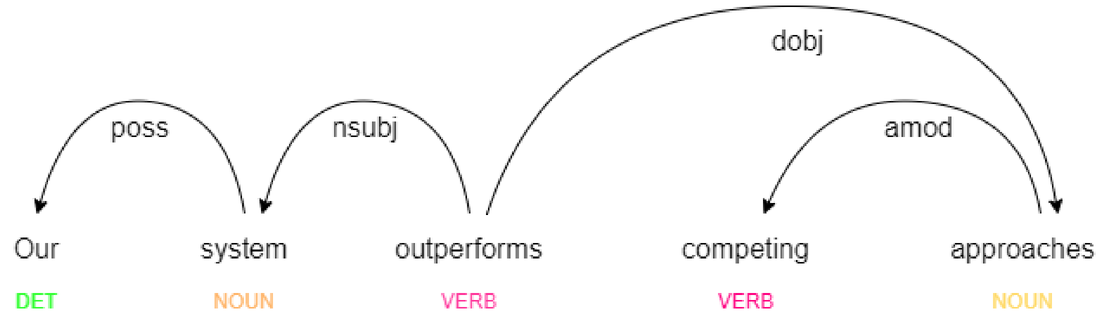

FIGURE 3.2: Dependency Structure

1. poss (system, our)
2. nsubj (outperforms, system)
3. amod (approaches, competing)
4. dobj (outperforms, approaches)

Finding dependency relationships are very helpful for citation analysis, therefore, scholars have focused on utilizing nsubj (nominal subject), advmod (adverb modifier) and amod (adjectival modifier) information in their systems [5, 42, 43]. These sorts of tags are also indicators of subjectivity in the statement of the sentences. From these references we got motivated to use the dependency structures of long-distance relationship between words in a sentence.

3.3.4 Window Based Negation

Negations are very important in linguistics because they have an effect on the polarities of other words. Negations contain terms like no, not, shouldn't, etc. Moreover, in case of negation in a sentence, it becomes necessary to identify words sequence affected by this term. There has been a lot of work which deals with negation and its scope in the Citations' classification. We used a negation list which contained 31 terms (no, not, rather, could not, was not, did not, would not, should not, were not, do not, does not, have not, has not, won't, wont, hadn't, never, none, nobody, nothing, neither, nor, nowhere, isn't, can't, cannot, mustn't, mightn't, shan't, without, needn't) [42]. We have detected the negation terms by using negation list and dependency tree. For the scope of negation, we have followed the window-based approach [5, 41]. We have used a negation window of 7 words for citations classification. Altogether words within a seven-word range of any terms of negation are suffixed through a token-neg to segregate between them and further versions.

3.4 Selection of Appropriate Rules

For citation polarity analysis, we developed feature extraction methods for extracting information from citation texts in order to increase the accuracy of the polarity. Following the completion of two tests on annotated data sets. It is discovered that the verbs, adverbs, nouns, and adjectives were the most useful parts of speech. So, these POS were useful in determining the reasoning class of citations. The data from the literature review was then used to identify a few key features that performed well in text classification. Parts of speech in this category include nouns, verbs, adverbs, and adjectives. We have selected a set of rules to extract different types of features from the citation, from all rules experimented depending upon the results. Table 3.2 represents all the rules with their results that have been experimented.

Table 3.1: All Rules

Sr #	**Rules**	**Results**	**Status**
1	Pick seven words after negation clause. Do not consider stop words.	86%	Selected
2	Pick all verbs form text.	80%	Not Selected
3	If a noun occurs immediate after advmod dependency, then pick noun.	88%	Selected
4	Pick all nouns as a feature form the text.	78%	Not Selected
5	If a ADJ occurs immediately after the verb or a verb hold (conj) dependency, then pick both verb and ADJ as a bigram.	87%	Selected
6	If adverb (ADV) occurs after the punctuation mark, then pick the ADV.	50%	Not Selected
7	Pick all ADJ as a feature form the text.	69%	Not Selected
8	If a word holds adjectival modifier (amod) dependency with a word, then pick a unigram and bigram.	89%	Selected
9	If subordinating conjunction (sconj) label words occur at the start of the sentence, then pick it.	62%	Not Selected
10	If adposition (ADP) label words occur before the punctuation mark, then pick the ADP.	67%	Not Selected
11	If adj or adverb occur immediate after nsubj then pick ADV or ADJ.	87%	Selected
12	Pick all adverbs from the text.	78%	Not Selected
13	If adverb (ADV) occurs before nominal subject (nsubj) dependency, then pick ADV.	67%	Not Selected
14	If adjective (ADJ) occurs after nominal subject (nsubj) dependency, then pick ADJ.	73%	Not Selected

Table: 3.1 - Continued from Previous Page

Sr #	Rules	Results	Status
15	Pick 3 words after negation clause. Do not consider stop words.	75%	Not Selected

3.5 Formulation of Feature Selection Rules

We have made rules that can pick only negated words from the text, but this will not show the results up to the mark. We followed Athar's [5] strategy to pick negation terms that are suffixed with a token neg to distinguish them from other text phrases versions. Negatives include no, not, shouldn't, and so on. Furthermore, when there is a negation in a sentence, it is necessary to determine which words in the sentence's sequence are affected by this term. Much work has been done on negation and its place in the Citations classification system.

A negation list with 31 terms (no, not, rather, could not, was not, did not, would not, should not, were not, do not, does not, has not, has not, will not, will not have, has not, won't, won't have) was used [42]. The negation terms were identified using a negation list and a dependency tree. To determine the scope of negation, we have used the window-based approach [5] [41]. A negation window of seven words was employed for the classification of citations. To distinguish them from other versions of the same word, all words falling within a seven-word range of any terms of negation are suffixed with a token-neg. By considering all these strategies, our rule gives good results. Similarly, we have simple rules for obtaining nouns from different citations, but that was not a worthwhile effort. Then we decided to make rules by considering dependency features. The first set of these characteristics includes typed dependency structures [35], which define the grammatical links that exist between different words. We want to capture the long-distance links that exist between words and phrases.

For citation analysis, dependency relationships are extremely useful; hence, several researchers have concentrated on including nsubj (nominal subject), advmod (adverb modifier), and amod (adjectival modifier) information into their systems. These kinds of tags are also evidence of the presence of subjectivity in the sentences' statements of fact. Using dependency structures, we will be able to capture the long-distance interaction between words. Bi-tagged features also employed that were obtained by POS tagging.

A technique for extracting sentiment rich bi-word features in which each part is either an adjective or an adverb, for example, adjective-noun, adverb-adjective, noun-adjective, and adverb-verb. Additionally, we have discovered that the verb (verb-noun, verb-adjective, adjective verb, and adverb-verb) can provide useful reasoning information for citation classification. N-grams features also employed in rules formulation process. Bigrams that use a lot of adjectives and adverbs are thought to be more meaningful than other types. It's also a good idea to focus on the negation words themselves. For our experiments, the lengths of the n-grams range from one to two.

3.5.1 Flowcharts of Formulation of Feature Selection Rules

This diagrammatic representation depicts a process for resolving a specific issue. Below are the diagrams which has the full description of rules formulation.

3.5.1.1 Formulation of Rule 1

In linguistics, negations are critical because they affect the polarity of other words. Negations include the words no, not, should not, and so on. Additionally, when a phrase contains negation, it becomes important to determine the words sequence impacted by this term. There has been a great deal of study done on negation and its application to the Citations categorization. Figure 3.4 shows the complete process of the implemented algorithm. The Spacy library is employed for the tagging process. The below figure shows all processes of rule that how this rule is working and extract desired information form the citation sentences. A negation list and a dependency tree were used to identify the negation words. To handle negations, we need tagged citation sentences (POS tagged). These citation sentences served as input to perform different sorts of operations, like checking whether a negation occurred or not.

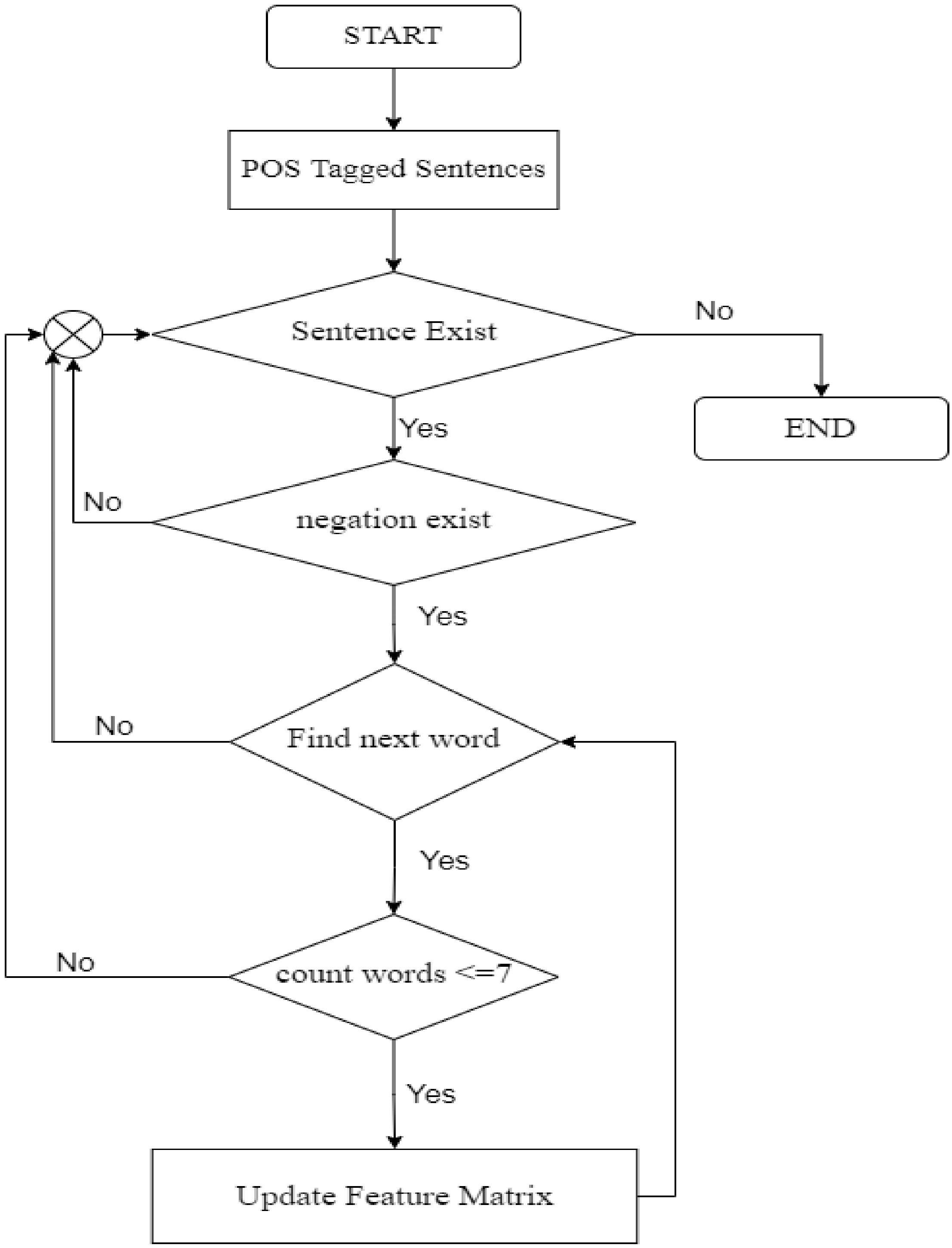

FIGURE 3.3: Rule 1 Formulation

To generate a feature matrix, we have to form a dictionary of unique words by using the feature selection rules from these citation texts, e.g. (VERB: work, donot, neg:work, VERB:use, ADJ:similar, similar to). Then to make the vector of the citation text, the terms of the citation text are matched with dictionary words. If the term matched, placed '1' in that index if not, place '0'. So similar case is here if there is found to be any negation dependency and the citation text matches with a dictionary word, then place '1' in that index and if the citation text did not match with a dictionary word place '0' in that index. There is another important condition present, like picking up to 7 words after a negation term, e.g. Consider the sentence, as for the tagger of Ratnaparkhi (1996) cannot tag a word lattice, so we cannot back off this tagging. If we pick upto seven words after negation, then those words will be (cannot, not tag, not word, not lattice, not back, not off, not tag). We classified citations using a negation window of seven words. All words falling within a seven-word range of negation keywords are suffixed with a token-neg.

3.5.1.2 Formulation of Rule 2

Dependency characteristics indicate the grammatical relationship that exists between the two terms. In the dependency structure, each feature indicates a binary connection between a head word and a dependent word. Similarly, we have simple rules for obtaining nouns from different citations, but that was not a worthwhile effort. Then we decided to make rules by considering dependency features.

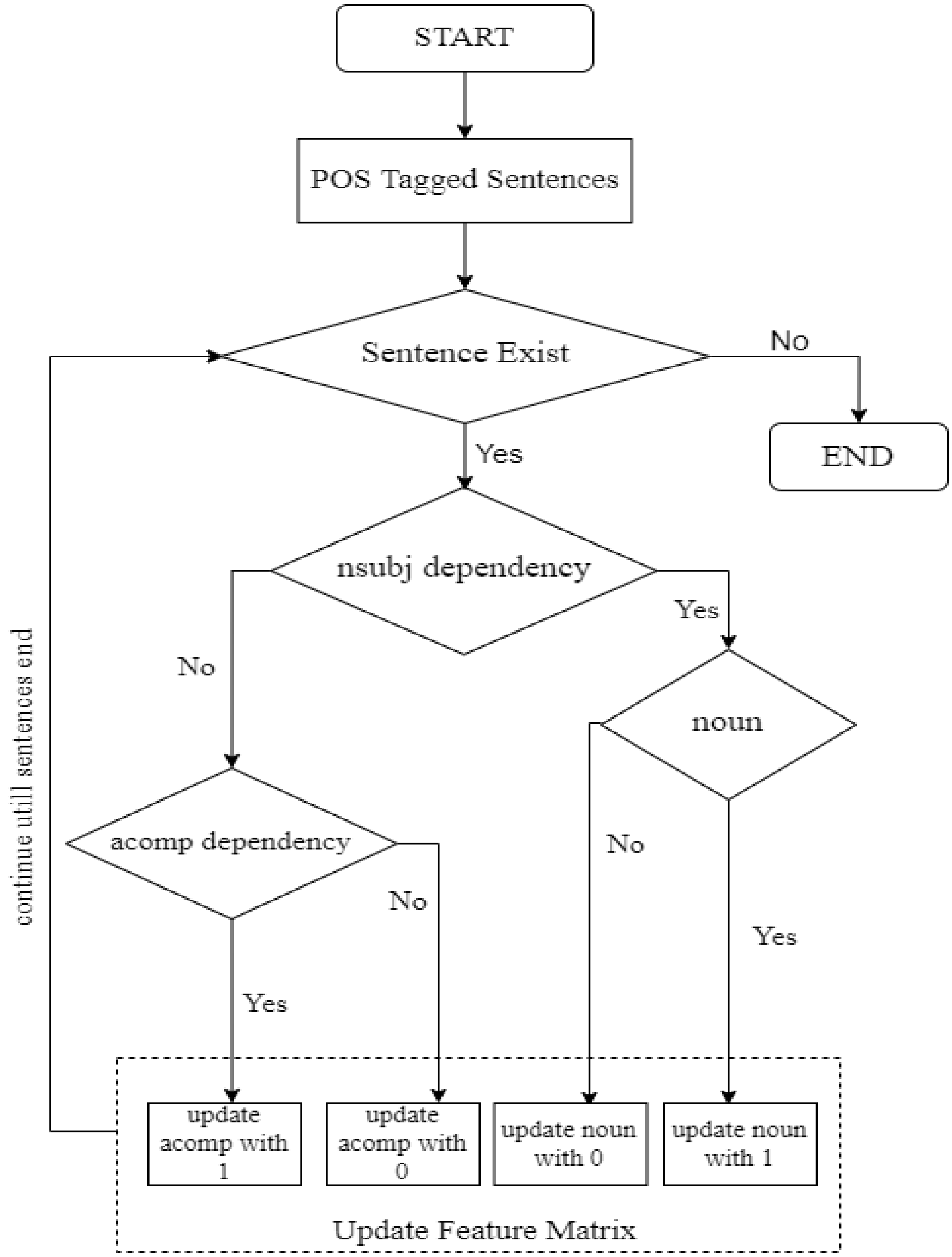

Figure 3.4: Rule 2 Formulation

The first set of these characteristics includes typed dependency structures, which define the grammatical links that exist between different words. We want to capture the long-distance links that exist between words and phrases. As you can see in Figure 3.2 that for citation analysis, dependency relationships are extremely useful; hence, several researchers have concentrated on including nsubj (nominal subject), advmod (adverb modifier), and amod (adjectival modifier) information into their systems. These kinds of tags are also evidence of the presence of subjectivity in the sentences' statements of fact. Using dependency structures, we will be able to capture the long-distance interaction between words. As you can see in the flowchart, we need sentences that have been labeled (POS tagged). Then the next step is to find a nominal subject (nsubj) dependency.

After that, we have to find the noun from that sentence. If a noun is discovered following nominal subject dependency, then the feature matrix must be updated accordingly. Meanwhile, from POS tagged sentences, we have to check whether an adjective complement (acomp) dependency exists or not. If a dependency exists, we must update the feature matrix; accordingly, if no such dependency exists, we place a '0' value against a specific index. Repeat this procedure until the rules are applied to all sentences that are present in that dataset. By the formulation of these sort of rules we got good results.

3.5.1.3 Formulation of Rule 3

The term "dependency characteristics" refers to the grammatical link between the two terms. Each feature in the dependency structure denotes a binary relationship between a head word and a dependent word. Additionally, we used bi-tagged features obtained through POS tagging. An approach for extracting sentiment-rich bi-word features composed of either adjectives or adverbs, such as adjective-noun, adverb-adjective, noun-adjective, and adverb-verb.

Furthermore, we discovered that the verb (verb-noun, verb-adjective, adjective verb, and adverb-verb) can provide useful reasoning information for citation classification. As illustrated in the flowchart, we require labeled sentences (POS tagged). Then we must determine whether adposition (preposition and postposition) or conjunction (conj) dependency exists or not. If any dependency exists, the next step is to identify the adjective (ADJ) or adverb (ADV).

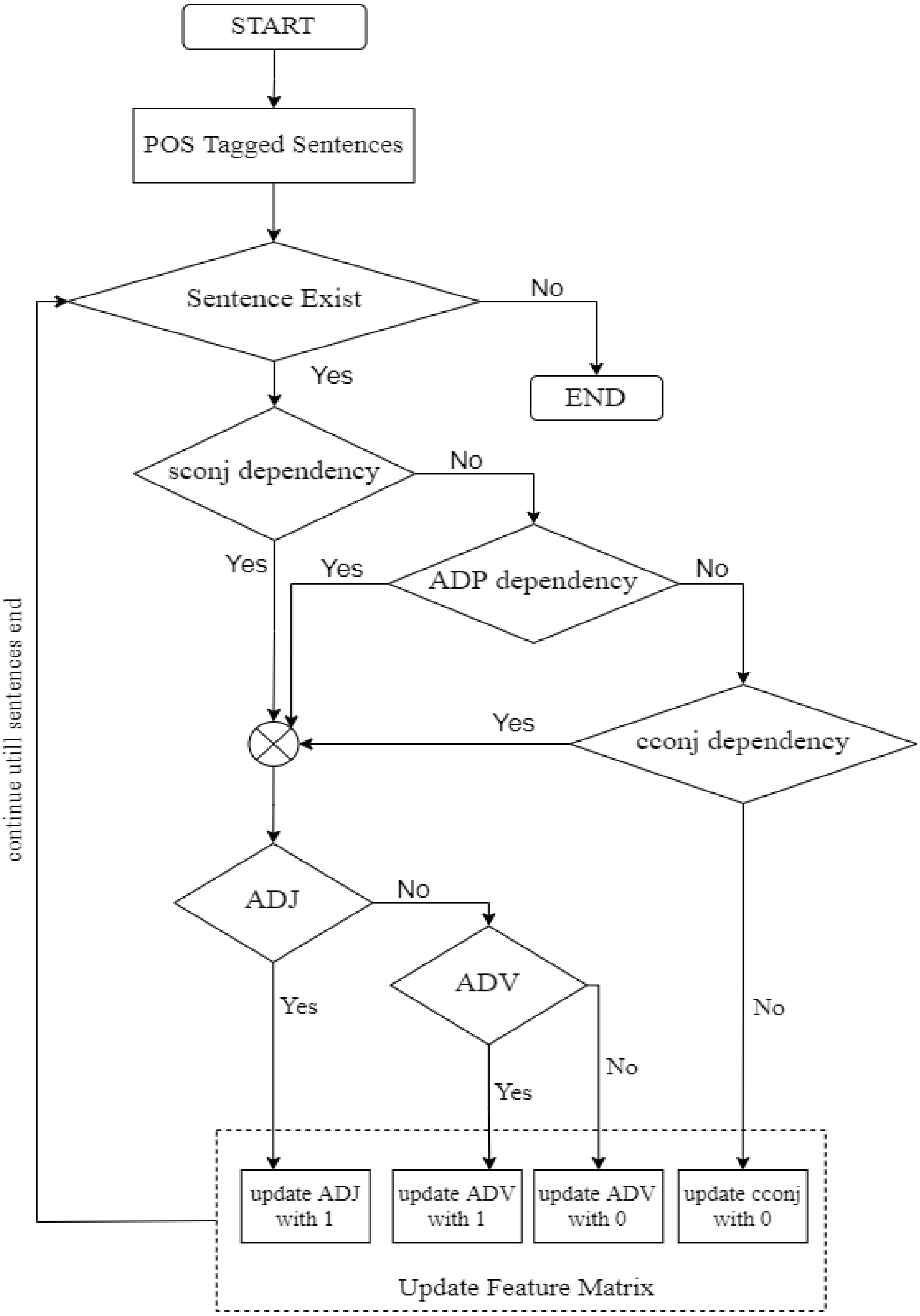

FIGURE 3.5: Rule 3 Formulation

There are two possibilities here, such as whether we found (ADJ or ADV) or not. If we find it, then the next step will be to place a '1' value against a specific index. If we did not find (ADJ or ADV), we must update our feature matrix by assigning a '0' value to each index. If adposition (preposition and postposition) or conjunction (conj) dependency does not exist, then we must update our feature matrix by assigning a '0' value. Continue in this manner until all of the sentences in the dataset that are contained inside it have had the rules applied to them all. As a result of the construction of these sorts of regulations, we were able to achieve good results.

3.5.1.4 Formulation of Rule 4

The grammatical relationship that exists between the two terms is indicated by their dependency qualities. Each feature in the dependency structure denotes a binary relationship between a head word and a dependent word between two words. As an example, we have basic criteria for extracting nouns from various citations, but it was not a profitable endeavor. As a result, we decided to develop rules that took dependent characteristics into consideration. The first of these traits is the presence of typed dependency structures, which define the grammatical ties that exist between various words. Long-distance connections between words and sentences are what we're looking for to capture. As you can see in Figure 3.2 that for citation analysis, dependency relationships are extremely useful; hence, several researchers have concentrated on including nsubj (nominal subject), advmod (adverb modifier), and amod (adjectival modifier) information into their systems. These kinds of tags are also evidence of the presence of subjectivity in the sentences' statements of fact. Using dependency structures, we will be able to capture the long-distance interaction between words. As you can see in the flowchart, we need sentences that have been labeled (POS tagged). Then the next step is to find a nominal subject (nsubj) dependency. There are two possibilities here, such as whether we found a nominal subject (nsubj) or not. If we did not find a nsubj dependency, we will return to the "POS Tagged Sentences" step and select the next sentence, repeating this process until we find an nsubj dependency. After successfully determining nsubj dependency, our next step is to find an adjective (ADJ) or adverb (ADV) from that sentence. There are two possibilities here, such as whether we found (ADJ or ADV) or not. If we find it, then the next step will be to place a '1' value against a specific index. If we did not find (ADJ or ADV), we must update our feature matrix by assigning a '0' value to each index. Continue in this manner until all of the sentences in the dataset that are contained inside it have had the rules applied to them all. As a result of the construction of these sorts of regulations, we were able to achieve favorable results.

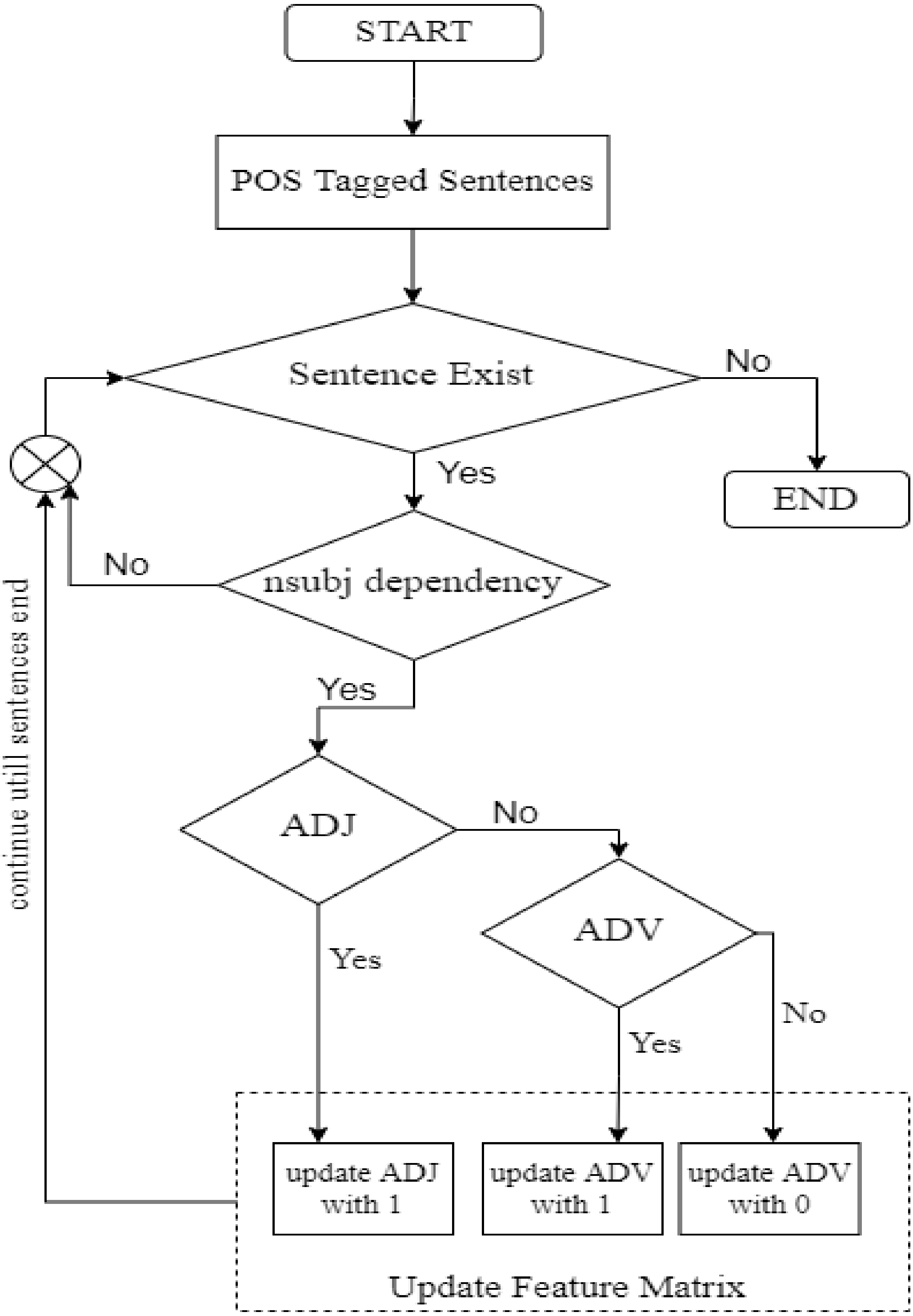

Figure 3.6: Rule 4 Formulation

3.5.1.5 Formulation of Rule 5

The grammatical link that exists between the two words is represented by their dependence characteristics. Each feature in the dependency structure represents a binary relationship between a head word and a dependent word between two terms. As an example, we have basic criteria for extracting nouns from diverse citations, but it was not a lucrative undertaking. As a result, we decided to create rules that took dependent qualities into mind. The first of these features is the presence of typed dependency structures, which specify the grammatical relationships that exist between distinct words.

Long-distance relationships between words and phrases are what we're searching for to capture. As you can see in the flowchart, we need sentences that have been labeled (POS tagged). Then the next step is to find a nominal subject (nsubj) or passive nominal subject (nsubjpass) dependency. There are two possibilities here, such as whether we found a nominal subject (nsubj) or not. If we did not find a nsubj dependency, we will return to the "POS Tagged Sentences" step and select the next sentence, repeating this process until we find a nsubj or nsubjpass dependency. After successfully determining nsubj or nsubjpass dependency, our next step is to find a verb from that sentence. As you can see in Figure 3.2 that for citation analysis, dependency relationships are extremely useful; hence, several researchers have concentrated on including nsubj (nominal subject), advmod (adverb modifier), and amod (adjectival modifier) information into their systems.

There are two possibilities here, such as whether we found (verb) or not. If we find it, then the next step will be to place a '1' value against a specific index. If we did not find (verb), we must update our feature matrix by assigning a '0' value to each index. Continue in this manner until all of the sentences in the dataset that are contained inside it have had the rules applied to them all. We have to check whether an adjective complement (amod) dependency exists or not. If a dependency exists, we must update the feature matrix; accordingly, if no such dependency exists, we place a '0' value against a specific index. As a result of the construction of these sorts of regulations, we were able to achieve favorable results.

Those results will be helpful to extract important features from the citations. Important features will then used to predict context classes. Those context classes are positve, negative and neutral.

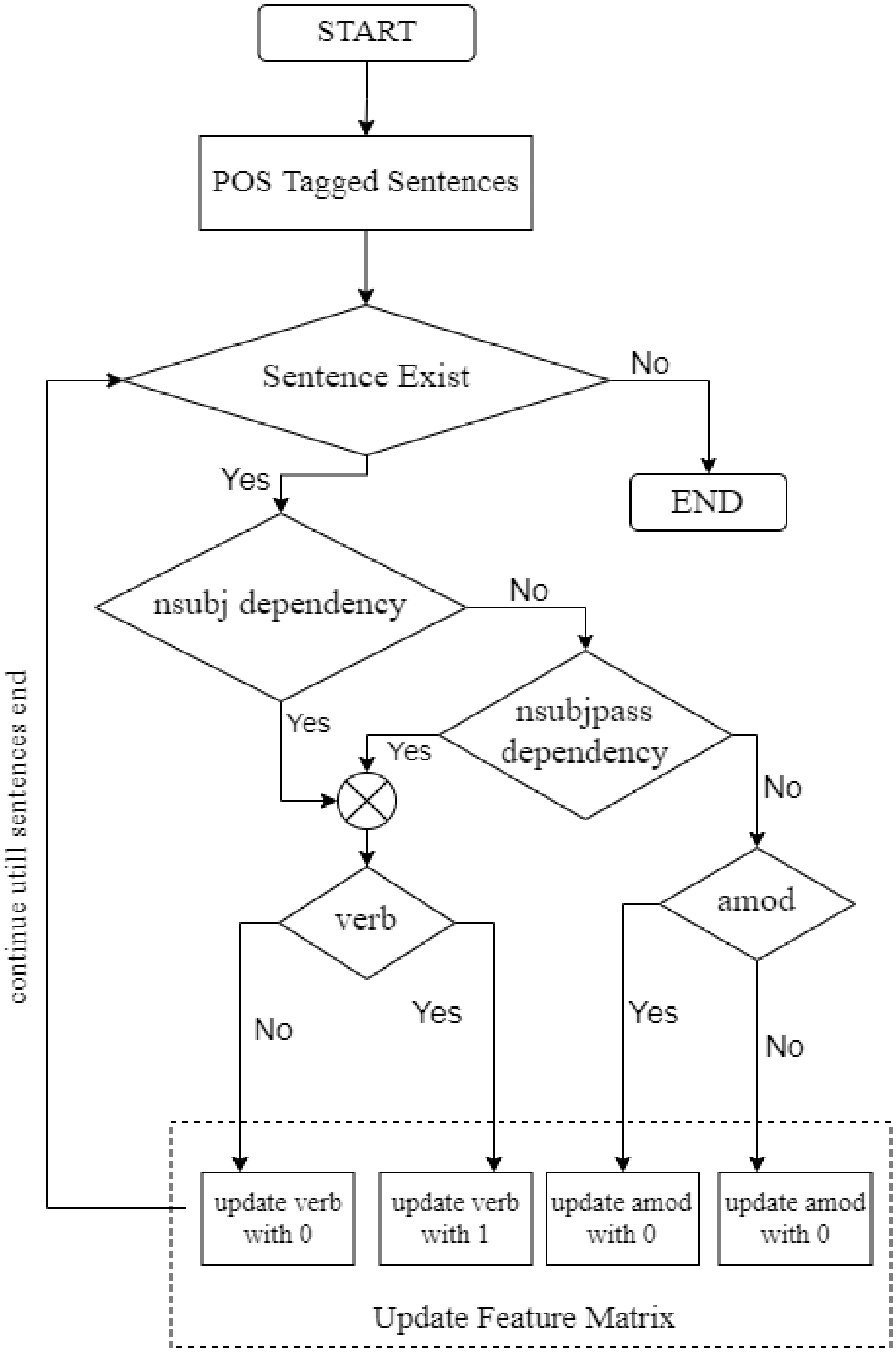

Figure 3.7: Rule 5 Formulation

3.6 Feature Selection Rules

We devised feature selection rules to extract important features from citation texts to improve the accuracy of citations polarity analysis. These rules have been prepared by conducting two experiments on annotated data set. The most valuable part of speech was verbs, adverbs, nouns, and adjectives. Thus, these POS helped to figure out the reasoning class of citations. Secondly, the literature review identified some important parts of speech that performed well in text classification. These parts of speech include nouns, verbs, adverbs, and adjectives. Experiments have been performed on data set with these important parts of speech. After analyzing the results, some important parts of speech are more beneficial than other parts of speech. In these datasets, these parts of speech include verbs, adverbs, nouns, and adjectives, which informs well about the relevant context class. Classification upon these datasets play an important role to find the context classes and those context classes are positive, negative and neutral. In this way, feature selection rules are devised to extract important features from citation texts. These feature selection rules are given in Table 3.2

Table 3.2: Features Extraction Rules

Sr #	Rules	Examples	Features
1	If a word holds adjectival modifier (amod) dependency with a word then pick a unigram and bigram	(Cutting et al., 1992) testified extraordinary results (96% on the Brown corpus) for unsupervised POS tagging using Hidden Markov Models (HMMs) by using hand-built tag dictionaries and equivalence classes.	AMOD: extraordinary-results ADJ: extraordinary NOUN: results

Table: 3.2 - Continued from Previous Page

Sr #	**Rules**	**Examples**	**Features**
2	If a noun occur immediate after advmod dependency, then pick noun.	Experiments, by using 4 algorithms and through visualization techniques, revealed that clustering is a worthless effort for paraphrase corpora construction, contrary to the literature claims (Barzilay & Lee, 2003)..	ADJ: worthless effort
3	If adj or adverb occur immediate after nsubj then pick ADV or ADJ.	There are however other similarity metrics (e.g. BLEU (Papineni et al., 2002)) which could be used equally well.	ADJ: equal
4	Pick seven words after negation clause or contraction clue before the punctuation mark. Do not consider stop words.	As the tagger of Ratnaparkhi (1996) cannot tag a word lattice, we cannot back off to this tagging.	Negated words: cannot, not tag, not word, not lattice, not back, not off, not tag

Table: 3.2 - Continued from Previous Page

Sr #	Rules	Examples	Features
5	If a ADJ occurs immediately after the verb or a verb hold (conj) dependency, then pick both verb and ADJ as a bigram.	Introduction Statistical phrase-based systems (Och and Ney, 2004; Koehn et al., 2003) have consistently delivered state-of -the-art performance in recent machine translation evaluations, yet these systems remain weak at handling word order changes.	CONJ: remain-weak ADJ: weak VERB: remain

3.7 Vectorization

Numeric vectors are quite often used as input in machine learning algorithms. However, before we conduct any operation on text, we must convert each citation text to a numeric vector which will result into a feature matrix. In feature matrix, each word is converted into a binary value 1 or 0, which indicate the word appear in a citation text or not. Several types of features that capture the characteristics of citation sentences are generated by feature extraction module and then serve as the input for the classifiers.

Let us consider an example to understand the working of feature matrix. Assume there are five citation texts as given below:

1. Sublanguage techniques do not work.

2. The model we use is similar to (Ratnaparkhi, 1996).
3. The systems remain weak at handling word changes.
4. Clustering is a worthless effort for paraphrase corpora construction.
5. There are however other similarity metrics which could be used equally well.

First we have to form a dictionary of unique words from these citation texts such as: (VERB: work, donot, neg:work, VERB:use, ADJ:similar, n similar to) etc. Then to make the vector of the first citation text, the terms of the citation text are matched with dictionary words. If term matched placed '1' in that index if not, then placed '0'.

Table 3.3 lists all the features and their associated serial numbers, whereas Table 3.4 shows the feature matrix.

Table 3.3: Features

Sr.#	**Features**
F1	VERB: work
F2	do-not
F3	neg: work
F4	VERB: use
F5	ADJ: similar
F6	NOUN: system
F7	AJD: weak
F8	VERB: handling
F9	NOUN: clustering
F10	ADJ: worthless
F11	NOUN: metrics

TABLE 3.4: Vectorization

Features / Citations	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Citation 1	1	1	1	0	0	0	0	0	0	0
Citation 2	0	0	0	1	1	0	0	0	0	0
Citation 3	0	0	0	0	0	1	1	1	0	0
Citation 4	0	0	0	0	0	0	0	0	1	1
Citation 5	0	0	0	1	0	0	0	0	0	0

3.8 Classification Techniques

Majority of the techniques surveyed (more than 70%) uses NB, RF, SVM, and J48 classifiers. We will also use these classifiers for the evaluation of features selection rules.

3.8.1 Naive Bayes (NB)

The NB classifier [17] works on the basis of "probabilistic classifiers" based on applying Bayes' theorem with strong independence assumptions between the features. It shows better performance with multi-class problems and perform better in text classification. Moreover, the model is simple to create, and particularly useful for huge datasets.

3.8.2 Random Forest (RF)

The Random Forest (RF) classifier [44] are suitable for dealing with the high dimensional noisy data in text classification. An RF model comprises a set of decision trees each of which is trained using random subsets of features. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing categorical variables as in the case of classification.

It performs better results for classification problems. This approach is ensemble based which is better than a single decision tree approach, and by averaging the result it reduces the over fitting.

3.8.3 Support Vector Machine (SVM)

SVM [16] is one of the powerful and flexible learning-based algorithms which is commonly used for classification task on labeled data sets. SVM work by finding a line that best separates the data into the two groups. This is done using an optimization process that only considers those data instances in the training dataset that are closest to the line that best separates the classes. The instances are called support vectors, hence the name of the technique. In almost all problems of interest, a line cannot be drawn to neatly separate the classes, therefore a margin is added around the line to relax the constraint, allowing some instances to be misclassified but allowing a better result overall. SVM classifiers have excellent precision and function well with high dimensional space. Basically, SVM classifiers use subset of training points thus uses very less memory. SVMs work well on small as well as high dimensional data spaces. It works effectively for high-dimensional datasets because of the fact that the complexity of the training dataset in SVM is generally characterized by the number of support vectors rather than the dimensionality.

3.8.4 J48

Quinlan's C4.5 [22] algorithm actualizes J48 to create a trimmed C4.5 decision tree. Every aspect of the information is split into minor subsets based on a decision. J48 looks at the standardized data gain that separates the information by choosing an attribute. To summarize, the attribute extreme standardized data gained is utilized. The algorithm returns the minor subsets. The split strategies stop if a subset has a similar class in all the instances. J48 develops a decision node utilizing the expected estimations of the class. J48 can help to make accurate predictions from the data. It deals with the problems of the numeric attributes. It also requires less data cleaning.

3.9 Evaluation

The objective of evaluation is to measure the relevance of particular features for the finding the sentiment of citations. If a particular rule to extract features give higher F1 than that rule is selected. For experimental purpose Weka tool is used for classification [45]. The classifiers selected for evaluation are SVM, RF, NB, and J48. We have used the training/testing data set in a 10-fold cross validation mode [46]. To calculate the results, precision, weighted-average precision, macro precision, recall, weighted-average recall, macro-recall, F1-score, weighted-average F1-score and macro-F were calculated.

Both corpora have multiple classes (positive, negative, and neutral), so instead of accuracy, we calculate macro-F to measure the system's overall performance. The evaluation parameters used for the single label multi class classification are given below:

$$Precision(P) = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{3.1}$$

$$Recall(R) = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3.2}$$

$$F1\_Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.3}$$

$$Weighted\_Avg(P) = \frac{1}{totalsamples}\sum_{i=1}^{n}((samplesofclass^{i}) * Pi) \tag{3.4}$$

$$Weighted\_Avg(R) = \frac{1}{totalsamples}\sum_{i=1}^{n}((samplesofclass^{i}) * Ri) \tag{3.5}$$

$$Weighted\_Avg(F1\_score) = 2 * \frac{Weighted\_Avg(P) * Weighted\_Avg(R)}{Weighted\_Avg(P) + Weighted\_Avg(R)} \tag{3.6}$$

$$Macro\_Precision = \frac{1}{n}\sum_{i=1}^{n} Pi \tag{3.7}$$

$$Macro\_Recall = \frac{1}{n}\sum_{i=1}^{n} Ri \tag{3.8}$$

$$Macro\_F = 2 * \frac{Macro\_Avg(p) * Macro\_Avg(R)}{Macro\_Avg(P) + Macro\_Avg(R)} \tag{3.9}$$

3.10 Tools

The tools and technical methodologies that were used for execution evaluating and calculation of the proposed approaches are given below:

- Spacy Library – Used to obtain the root of terms
- Excel – MS Office is used for design & diagrams for graphical table representations
- Python – Used for programming

3.11 Experimental Setup

All experiments are carried out on a window 10 equipped with four Intel(R) Core i5-3320M, 8 GB of RAM. For feature extraction we used PyCharm 2020.3.5 Official[1] with python 3.8. We also used Weka [45] tool for classification. For multiclass classification we used Support Vector Machine (SVM) with package LIBLINEAR [47] of version 1.9 and of batch size of 100. Although we have different kernels in SVM (Sigmoid, Gaussian Radial, Linear and Polynomial kernel) but we used linear kernel for solving multiclass classification problem. As we have three different classes, so linear kernel works relatively well when there is a clear margin of separation between classes. We have cost (C) value set to 1 because low value C tends to make decision surface smooth.

[1] https://www.jetbrains.com/pycharm/download/

Chapter 4

Results and Evaluation

This chapter presents the results, analyses them and compares with the other schemes' results.

4.1 Evaluation and Pre-Processing of Dataset

Two distinct datasets from the fields of computer science and bioinformatics are used for the studies. In our experiments, we used a specific version of the AAN data set. We have discussed about this data set in detail in the previous chapter. One of the corpora has been annotated by Athar [5] which contains 8,736 AAN citation sentences labeled as Citing Paper ID, Cited Paper ID, Citation Text, and one of the three sentiment classes which are positive, negative, and neutral. There are 829 positive citations, 280 negative and 7,627 neutral citations. There was a need to clean the dataset by removing stop words, extra brackets, extra spaces, and other similar characters. The 2$^{\text{nd}}$ dataset is derived from clinical study papers and contains citation sentences extracted from 285 randomly chosen publications [26]. The dataset comprises 4182 citation sentences arguing for biomedical study replication that were obtained from clinical trial papers. In total, 3172 citation sentences were found to be neutral, 702 positive and 308 negative citations. Table 4.1 summarizes both corpora's data and both datasets have been used in previous studies in the field of citation sentiment analysis.

Table 4.1: Descriptive Statistics of the Corpora

Serial Number	**No. of Citations**	**No. of Positive Citations**	**No. of Negative Citations**	**No. of Neutral Citations**
Dataset A	8736	829	280	7627
Dataset B	4182	702	308	3172

4.2 Important Parts of Speech Identification

There are several parts of speech in a citation text. We have identified such parts of speech from literature review, which performed well in text classification. These parts of speech include nouns, verbs, adverbs, and adjectives. We have trained NB, RF and SVM classifier on these features. In the experiments, we have trained our classifier one by one on parts of speech separately. First of all, on nouns, secondly on verbs, thirdly on adverbs and at the last on adjective. Afterwards, we trained these classifiers on these parts of speech collectively. We have used 10-fold cross validation to analyze the results of these classifiers.

Afterwards, we took three measurements named weighted-average precision, weighted-average recall, and weighted-average F1-score. With the help of these measurements, we analyzed the result accuracy ratio of the parts of speech shown in Figure 4.1. We have achieved maximum 82% weighted-average F1-score. After analyzing the results, we found that there are some parts of speech are helping more than other parts of speech. In this data set, these parts of speech include adjectives, verbs and adverbs which informs well about the relevant citations' functions. The result of SVM classifier is outperformed other classifiers. Furthermore, this experiment helped us a lot in the process of feature selection rules.

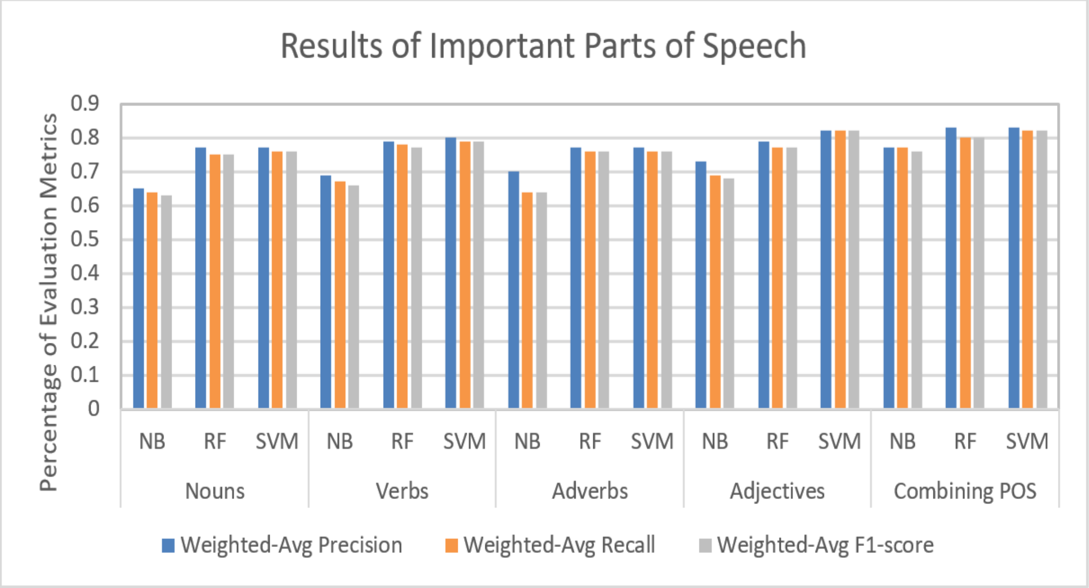

FIGURE 4.1: Results on POS

4.3 List of Features Extracted

Different rules have been formulated to extract important features from the citation text. In this thesis, five rules have been formulated to extract important features. Table 4.2 presents the extracted features from the citation sentences. Five different sentences are incorporated to extract features from them.

Table 4.2: List of Features Extracted

Features	**Citation 1**	**Citation 2**	**Citation 3**	**Citation 4**	**Citation 5**
cannot	1	0	0	0	0
neg:tag	1	0	0	0	0
neg:word	1	0	0	0	0
neg:	1	0	0	0	0
neg:lattice	1	0	0	0	0
neg:tagging	1	0	0	0	0
off_to	1	0	0	0	0
VERB:tag	1	0	0	0	1
contrary_to	0	1	0	0	0
VERB:reveal	0	1	0	0	0
amod:worthless-effort	0	1	0	0	0
ADJ:worthless	0	1	0	0	0
NOUN:effort	0	1	0	0	0
amod:contrary-effort	0	1	0	0	0
ADV:contrary	0	1	0	0	0
acomp:weak-remain	0	0	1	0	0
ADJ:weak	0	0	1	0	0
VERB:remain	0	0	1	0	0
weak_at	0	0	1	0	0
amod:statistical-system	0	0	1	0	0
ADJ:statistical	0	0	1	0	0
NOUN:system	0	0	1	0	0
amod:base-system	0	0	1	0	0
VERB:base	0	0	1	0	0
VERB:deliver	0	0	1	0	0
amod:recent-evaluation	0	0	1	0	0
ADJ:recent	0	0	1	0	0
NOUN:evaluation	0	0	1	0	0
NOUN:metric	0	0	0	1	0

Table: 4.2 - Continued from Previous Page

Features	Citation 1	Citation 2	Citation 3	Citation 4	Citation 5
amod:other-metric	0	0	0	1	0
ADJ:other	0	0	0	1	0
VERB:use	0	0	0	1	0
amod:extraordinary-result	0	0	0	0	1
ADJ:extraordinary	0	0	0	0	1
NOUN:result	0	0	0	0	1
amod:unsupervised-tag	0	0	0	0	1
ADJ:unsupervised	0	0	0	0	1
amod:Hidden-Models	0	0	0	0	1
PROPN:Hidden	0	0	0	0	1
PROPN:Models	0	0	0	0	1
amod:build-dictionary	0	0	0	0	1
VERB:build	0	0	0	0	1
NOUN:dictionary	0	0	0	0	1

4.4 Evaluation of Feature Extraction Rules

In the previous chapter, we have discussed feature extraction rules in detail. Several types of features that capture the characteristics of citation sentences are extracted by devised feature extraction rules are served as the inputs of automatic classifiers.

4.4.1 Evaluation of Rule 1

As we saw in the previous chapter, different combination of POS (noun-verb, noun-adjective, etc) gives good results as compared to other parts of speech (noun, verb, adverb, and adjective) results. We have trained our classifiers NB, RF, and SVM on important extracted features. We used a 10-fold cross validation approach to analyze the results of these classifiers. We have analyzed the results of rule one shown in Figure 4.2. We have achieved an 86% weighted-average F1-score. These results reveal that in classification while using rule one, the results have outperformed. Furthermore, the results of the SVM classifier have outperformed the results of the Naive Bayes (NB) and Random Forest (RF) classifiers.

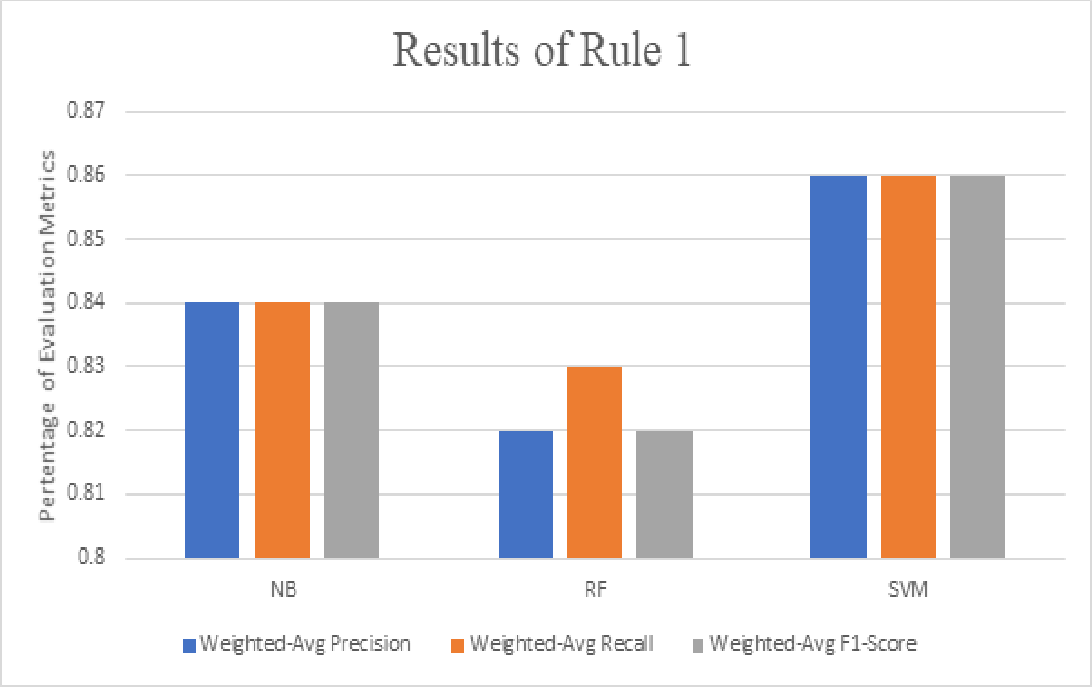

FIGURE 4.2: Results of Rule 1

4.4.2 Evaluation of Rule 2

Additionally, Rule two produces excellent results when compared to other parts of speech (noun, verb, adverb, and adjective). We trained our NB, RF, and SVM classifiers on significant extracted features. We analyzed the results of these classifiers using a 10-fold cross validation approach. We analyzed the rule two results depicted in Figure 4.3. We obtained an F1-score of 88 percent on a weighted average basis. These results indicate that when rule two is used in classification, the results outperformed. Additionally, the SVM classifier outperformed the Naive Bayes (NB) and Random Forest (RF) classifiers.

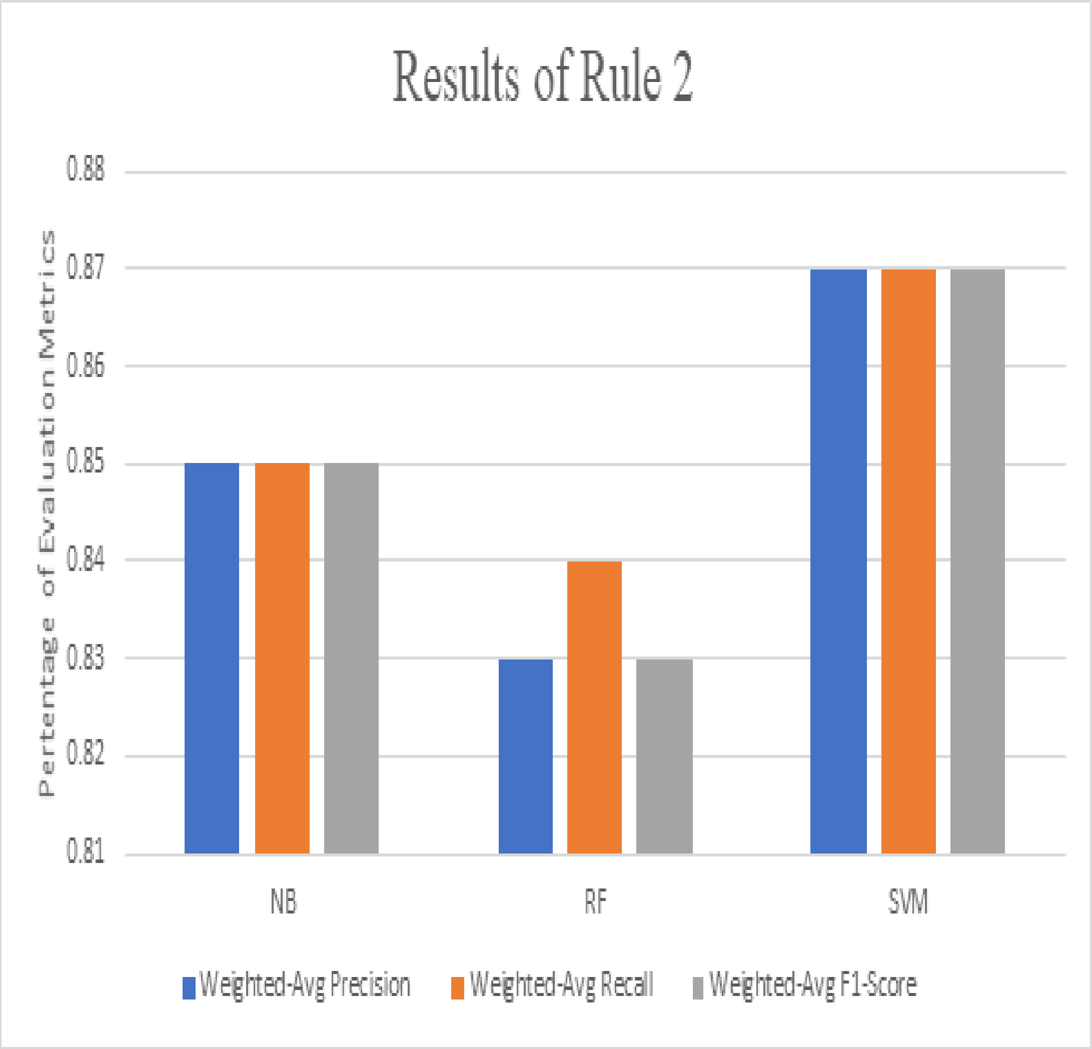

FIGURE 4.3: Results of Rule 2

4.4.3 Evaluation of Rule 3

Rule three also works well. Our NB, RF, and SVM classifiers were trained on extracted features. We used 10-fold cross validation to analyze these classifiers' results. Figure 4.4 shows the rule three results. Our weighted average F1-score was 87%. These results show that using rule three improves classification results. SVM outperformed Naive Bayes (NB) and Random Forest (RF) classifiers.

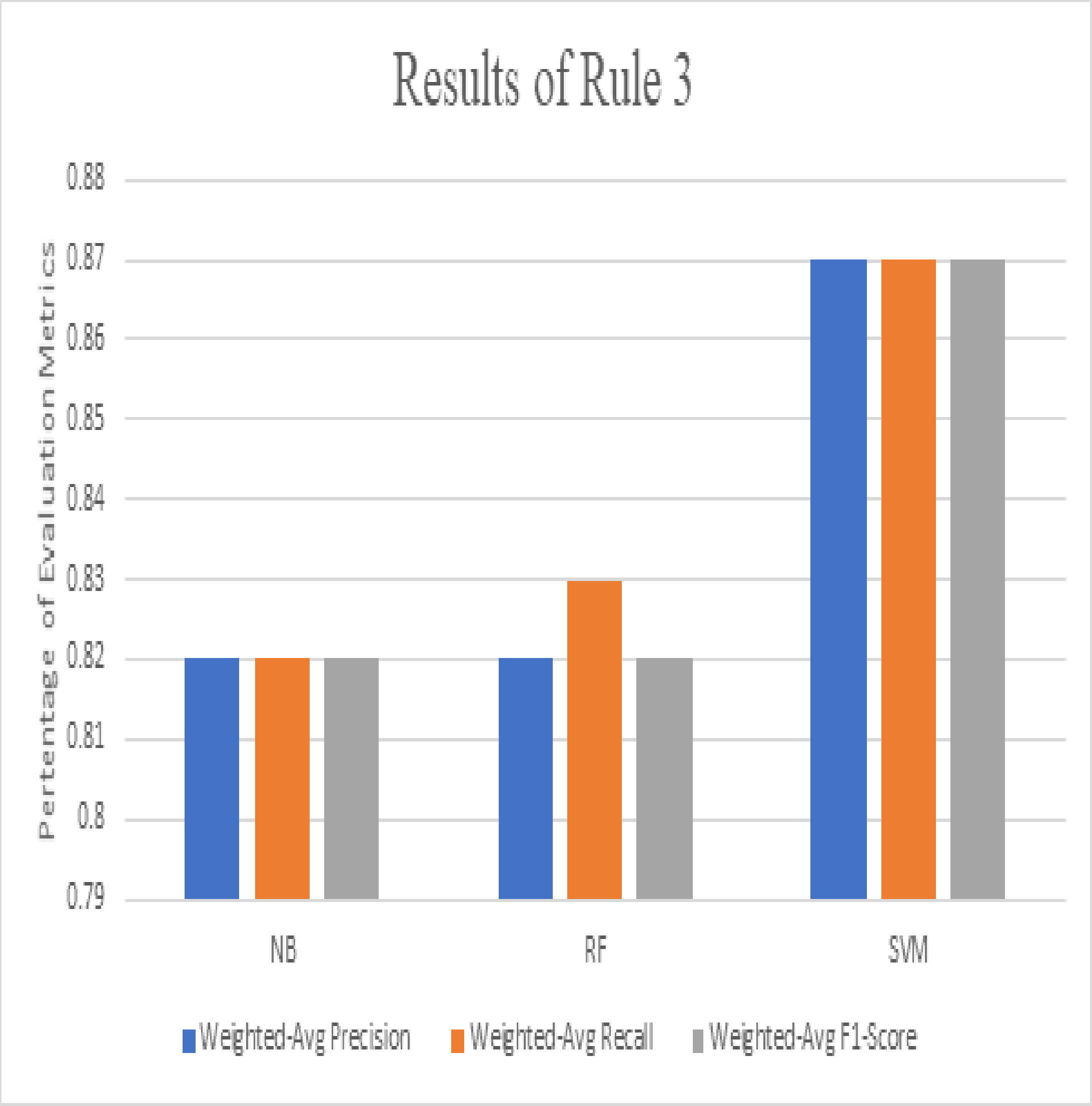

FIGURE 4.4: Results of Rule 3

4.4.4 Evaluation of Rule 4

Rule four is also successful in producing good outcomes. Our classifiers NB, RF, and SVM have all been trained using data from the extracted feature sets mentioned before. These classifiers were subjected to a 10-fold cross validation process in order to assess their performance. As you can see in Figure 4.5, the results of applying rule four were quite interesting. We have gotten an 87% on the weighted average of our F1 score. Using rule as a guide, these results show that the results outperformed. And the SVM classifier's results beat those of the Naive Bayes (NB), as well as those of the Random Forest (RF).

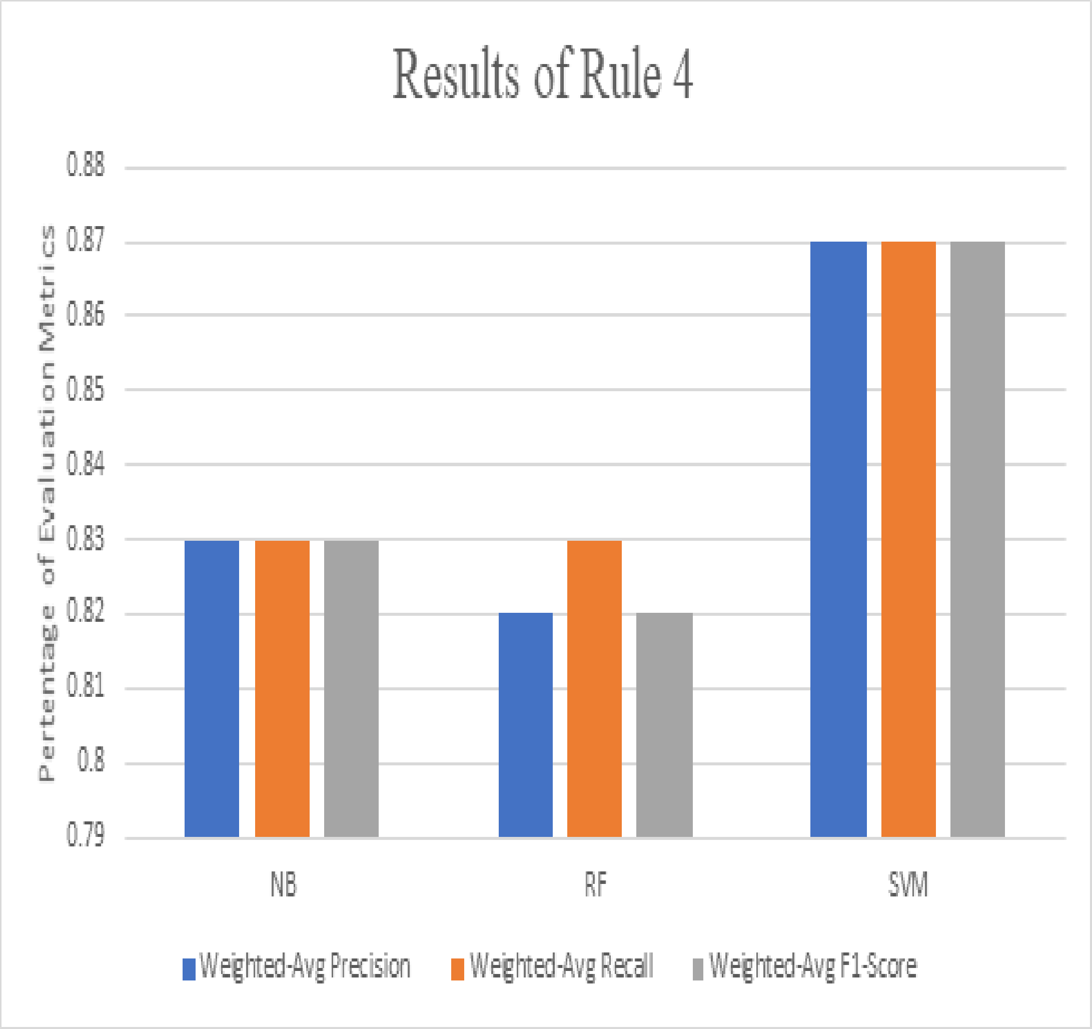

FIGURE 4.5: Results of Rule 4

4.4.5 Evaluation of Rule 5

Rule five is also very effective. Our NB, RF, and SVM classifiers were trained using the features that were extracted. The results of these classifiers were analyzed using a 10-fold cross validation procedure. The results of rule 1 are depicted in Figure 4.6. Our weighted average F1-score was 89 percent. These findings demonstrate that applying rule five improves classification results. The SVM classifier outperformed both the Naive Bayes (NB) and the Random Forest (RF) classifiers in terms of classification accuracy.

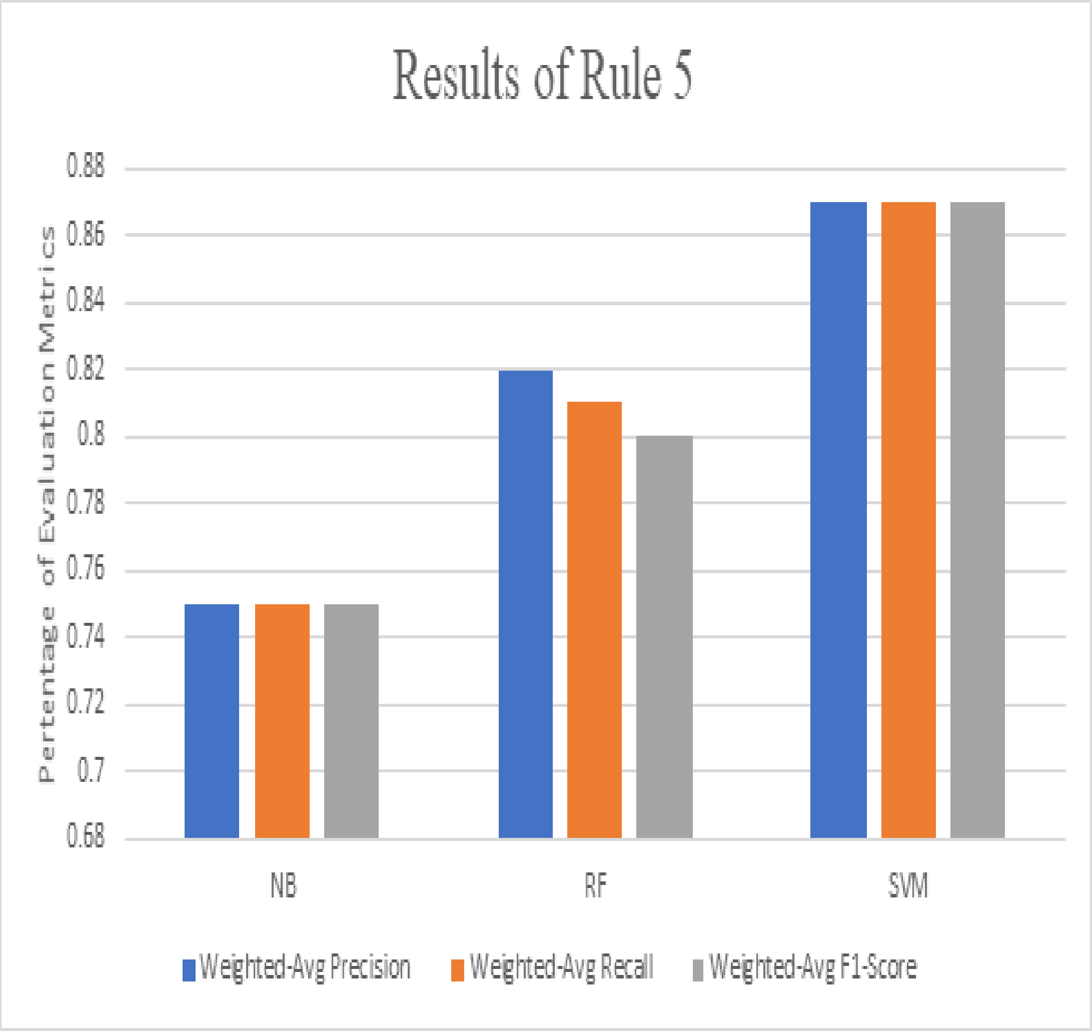

Figure 4.6: Results of Rule 5

4.4.6 List of Features Extracted

Different rules have been formulated to extract important features from the citation text. In this thesis, five rules have been formulated to extract important features. Table 4.2 presents the extracted features from the citation sentences. Five different sentences are incorporated to extract features from them.

4.4.7 Evaluation of Combined Rules on Athar's Dataset

As we have formulated different rules to extract important features from the citations, we have also performed experiments on each of the rules. It is clearly shown that all those rules give good results on Athar's [5] dataset. We have trained our classifiers NB, RF and SVM on important extracted features. Now we are going to train these classifiers on all those rules collectively. We used 10-fold cross authentication approach to analyze the results of these classifiers. For comparison purpose, we have used macro precision, macro recall and macro-F. With the help of these measures, we have analyzed the results of the feature extraction rules shown in figure 4.7. We have achieved 90% macro-F score. These results reveals that in classification while using rules have outperformed the results which were not using the rules. Furthermore, the result of SVM classifier has outperformed the results of Naive Bayes (NB), J48 and Random Forest (RF) classifiers.

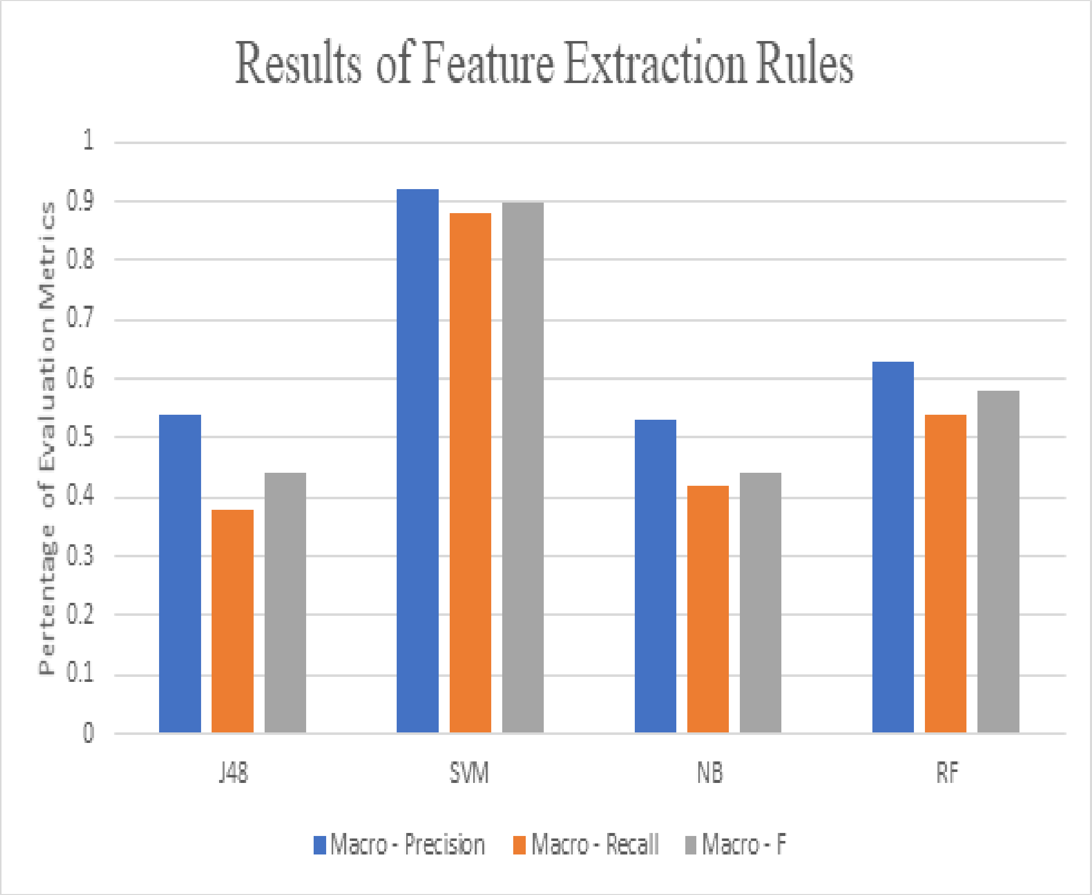

Figure 4.7: Results on Athar Dataset

4.4.8 Evaluation of Rules on Clinical Trials Dataset

Additionally, we conducted experiments using the Clinical Trials dataset. It is plain to see that all of those rules perform well on this dataset as well. On the basis of the extracted features, we trained our classifiers NB, RF, J48, and SVM. We analyzed the results of these classifiers using a tenfold cross validation approach. We compared macro precision, macro recall, and macro-F. We analyzed the results of the feature extraction rules shown in figure 4.8 using these measures. We achieved a macro-F score of 90%. These results demonstrate that when rules are used for classification, the results outperform. Additionally, the SVM classifier outperformed Naive Bayes (NB), J48, and Random Forest (RF) classifiers.

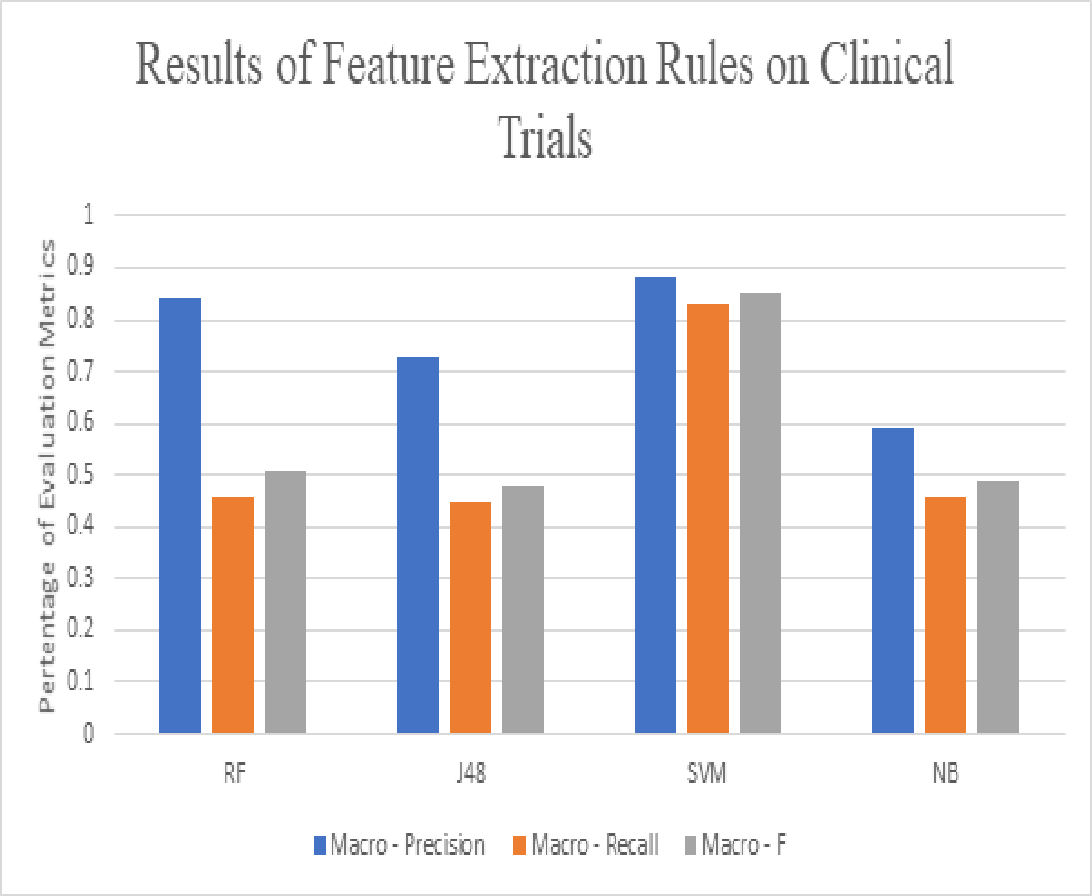

Figure 4.8: Results on Clinical Trails Dataset

4.5 Results of Rules on Combined Datasets

The citation analysis community has proposed various citation classification methods. This chapter's literature review portion describes many of these approaches. On the other hand, the best machine learning classifiers NB, RF, J48, and SVM were applied to improve the classification results. We compared macro precision, macro recall, and macro-F. We analyzed the results of the feature extraction rules shown in figure 4.9 using these measures. They analyzed the outputs of these classifiers using 10-fold cross validation. Both datasets Athar's [5] and Xu et al. [26] combined together for experiment purposes. SVM achieved good results as compared to other systems.

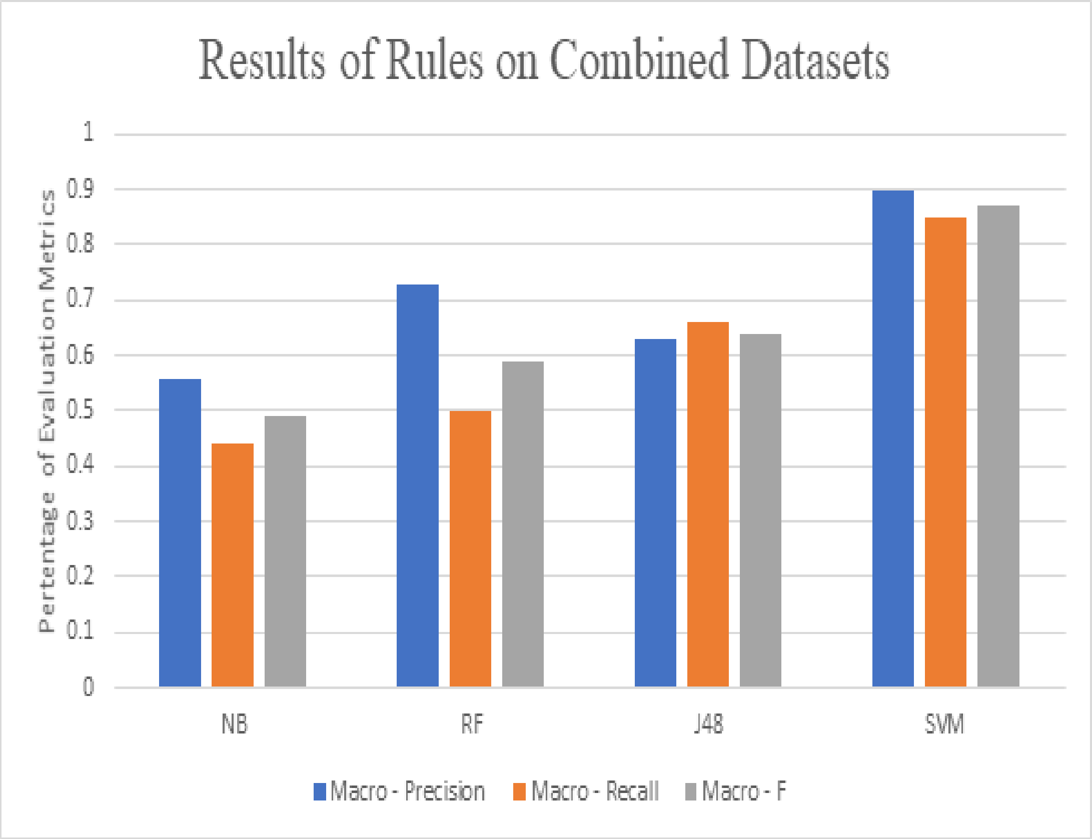

FIGURE 4.9: Results of Rules on Combined Datasets

4.6 Comparison with Other Systems on Athar's Dataset

The citation analysis community has proposed several methods for categorizing citations. The majority of these approaches made use of the various features and machine learning techniques outlined in the chapter on literature review. This thesis used feature selection criteria to extract critical characteristics from citation texts using the top machine learning classifiers NB, RF, J48, and SVM to improve the classification of citations. The output of these classifiers was analyzed using tenfold cross-validation. To make comparisons, macro precision, macro recall, and macro-F were used. We examined the classifiers' outputs using these measures and compared SVM findings to those of Jha et al. [7] and Mercier et al. [38]. These strategies were found to be more appropriate to the classifier and data set for our citation than the other approaches. Our proposed technique generated 90% macro–F, in contrast to the findings of Jha et al. [7] and Mercier et al. [38], who found that macro–F was 71% and 77 percent by performing citation's classification, respectively. Figure 4.10 illustrates the comparison of findings.

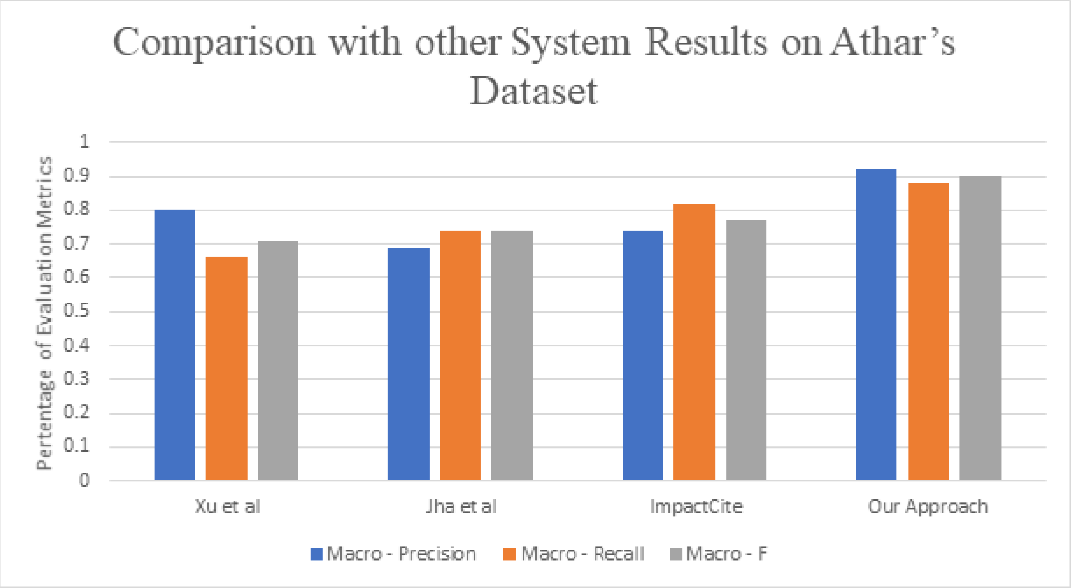

Figure 4.10: Comparison with Other System Results on Athar's Dataset

This thesis employed feature selection criteria to extract important features from citation texts and then classified them using the top machine learning classifiers NB, RF, J48, and SVM. To make comparisons, precision, recall, and F1 score were also used. We examined the classifiers outputs using these measures and compared SVM findings to those of Ikram et al. [31] and Yousif et al. [35]. Our proposed technique generated 95% F1 Score, in contrast to the findings Ikram et al. [31] and Yousif et al. [35], who found that F1 Score was 85% and 88 percent by performing citation's classification, respectively. Figure 4.11 illustrates the comparison of findings.

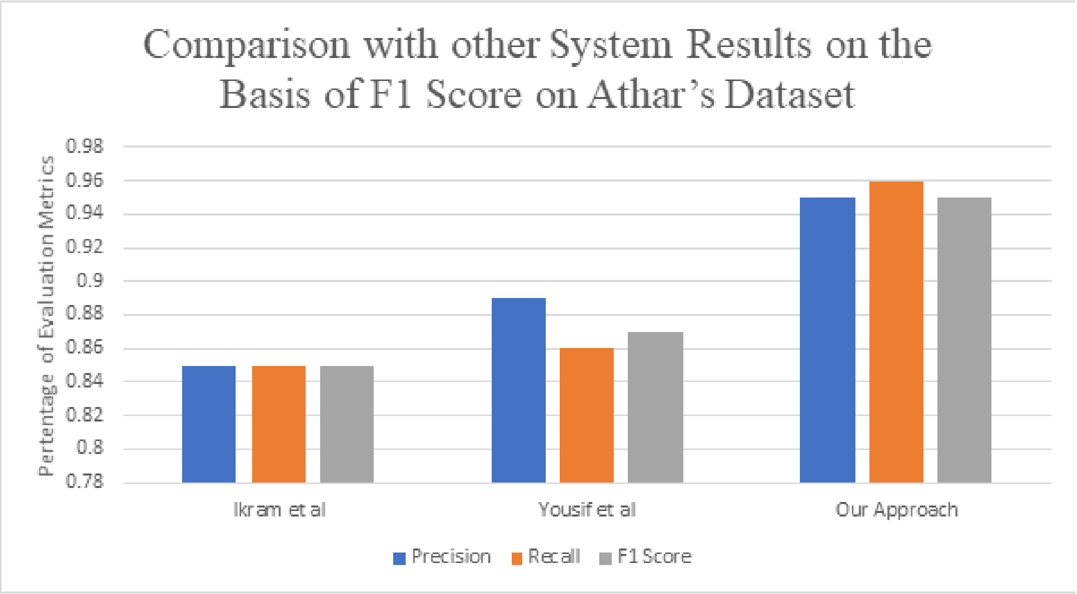

FIGURE 4.11: Comparison with Other System Results on Athar's Dataset on the Basis of F1-Score

4.7 Comparison with Other Systems on Clinical Trial Dataset

The citation analysis community has proposed several methods for categorizing citations. The majority of these approaches made use of the various features and machine learning techniques outlined in the chapter on literature review. The output of these classifiers was analyzed using tenfold cross-validation. For comparison purposes, macro precision, macro recall, and macro-F were used. We examined the classifiers' outputs using these measures and compared SVM findings to Xu et al. [26]. Our proposed technique generated 85% macro–F, in contrast to the findings of Xu et al. [26], who found that macro–F was 71% by performing citation's classification, respectively. Figure 4.12 illustrates the comparison of findings.

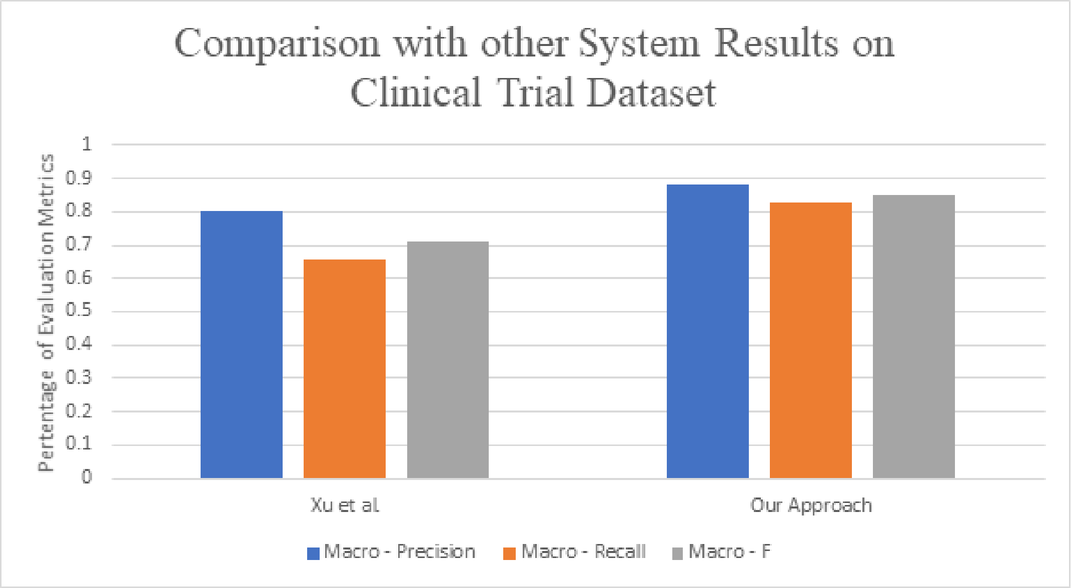

FIGURE 4.12: Comparison with Other System Results on Clinical Trial Dataset

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Citation sentiment analysis is an important task in order to know the reasons to cite a paper. There are two methodologies to detect sentiments; lexicon-based approach and corpus-based approach. A research gap to use hybrid approach was identified and a number of research questions were raised. It has been demonstrated that a hybrid approach using lexicon-based NLP rules to develop feature matrix and applying machine-learning algorithms on the extracted features, gives promising results.

Athar's data set, known as ACL anthology and Clinical Trial datasets were used in the experiments. First step was to pre-process the data in order to remove noise. The most important step in our research was the formulation of rules to select more relevant features for classification of citations into three sentiments: positive, natural, and negative. Several experiments were conducted to determine which features are critical for accurate classification. Our evaluation of the rules, most popular classifiers, the NB, the RF, the J48, and the SVM, were used.

In Athar's dataset two different parameters i.e. (macro-F and F1 score) were used for the evaluation of rules. The proposed approach produced 90% macro-F and 95% F1. In macro-F, the system produced 19% better results in comparison to Jha et al. [7] and 13% effective results respectively in comparison to Mercier et al.

[38]. Whereas in F1 score, the system generated 10% better results in comparison to Ikram et al. [31] and 7% better results when compared with Yousif et al. [35] respectively.

In the Clinical Trail dataset, the proposed technique generated 85% results where the method achieved 14% greater results during evaluations when compared to Xu et al. [26]. In conclusion, the rules to extract relevant features developed during this research demonstrated excellent performance.

5.2 Future Work

We have identified some research gaps that could be addressed in the future. These research gaps are described below:

1. The output of this research can be used in modern digital libraries to categorize the cited articles into three classes like positive, negative, and neutral.
2. Rules to extract citation polarity could be tested on a variety of datasets.

Bibliography

[1] H. Voos and K. S. Dagaev, “Are all citations equal? or, did we op. cit. your idem?.,” *Journal of Academic Librarianship*, vol. 1, no. 6, pp. 19–21, 1976.

[2] P. D. Turney, “Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews,” *arXiv preprint cs/0212032*, 2002.

[3] E. Garfield *et al.*, “Can citation indexing be automated,” in *Statistical association methods for mechanized documentation, symposium proceedings*, vol. 269, pp. 189–192, Washington, 1965.

[4] B.-A. Lipetz, “Improvement of the selectivity of citation indexes to science literature through inclusion of citation relationship indicators,” *American documentation*, vol. 16, no. 2, pp. 81–90, 1965.

[5] A. Athar, “Sentiment analysis of citations using sentence structure-based features,” in *Proceedings of the ACL 2011 student session*, pp. 81–87, 2011.

[6] I. Ihsan and M. A. Qadir, “Ccro: Citation’s context & reasons ontology,” *IEEE Access*, vol. 7, pp. 30423–30436, 2019.

[7] R. Jha, A.-A. Jbara, V. Qazvinian, and D. R. Radev, “Nlp-driven citation analysis for scientometrics,” *Natural Language Engineering*, vol. 23, no. 1, pp. 93–130, 2017.

[8] D. R. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara, “The acl anthology network corpus,” *Language Resources and Evaluation*, vol. 47, no. 4, pp. 919–944, 2013.

[9] M. J. Moravcsik and P. Murugesan, "Some results on the function and quality of citations," *Social studies of science*, vol. 5, no. 1, pp. 86–92, 1975.

[10] M. Garzone and R. E. Mercer, "Towards an automated citation classifier," in *Conference of the canadian society for computational studies of intelligence*, pp. 337–346, Springer, 2000.

[11] E. Brill, "Some advances in transformation-based part of speech tagging," *arXiv preprint cmp-lg/9406010*, 1994.

[12] S. Teufel, A. Siddharthan, and D. Tidhar, "Automatic classification of citation function," in *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 103–110, 2006.

[13] S. Teufel *et al.*, *Argumentative zoning: Information extraction from scientific text*. PhD thesis, Citeseer, 1999.

[14] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.

[15] K. Sugiyama, T. Kumar, M.-Y. Kan, and R. C. Tripathi, "Identifying citing sentences in research papers using supervised learning," in *2010 International Conference on Information Retrieval & Knowledge Management (CAMP)*, pp. 67–72, IEEE, 2010.

[16] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *the Journal of machine Learning research*, vol. 9, pp. 1871–1874, 2008.

[17] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," *arXiv preprint arXiv:1302.4964*, 2013.

[18] N. Tandon and A. Jain, "Citation context sentiment analysis for structured summarization of research papers," in *35th German conference on artificial intelligence*, vol. 98, Citeseer, 2012.

[19] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *European conference on machine learning*, pp. 406–417, Springer, 2007.

[20] A. Athar and S. Teufel, "Context-enhanced citation sentiment detection," in *Proceedings of the 2012 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies*, pp. 597–601, 2012.

[21] G. Parthasarathy and D. Tomar, "Sentiment analyzer: Analysis of journal citations from citation databases," in *2014 5th international conference-confluence the next generation information technology summit (confluence)*, pp. 923–928, IEEE, 2014.

[22] S. L. SALZBERG, "by j. ross quinlan. morgan kaufmann publishers, inc., 1993.," *Machine Learning*, vol. 1, p. 6, 1994.

[23] M. Hernández-Alvarez and J. M. Gómez, "Citation impact categorization: for scientific literature," in *2015 IEEE 18th International Conference on Computational Science and Engineering*, pp. 307–313, IEEE, 2015.

[24] B. H. Butt, M. Rafi, A. Jamal, R. S. U. Rehman, S. M. Z. Alam, and M. B. Alam, "Classification of research citations (crc)," *arXiv preprint arXiv:1506.08966*, 2015.

[25] S. Evert, T. Proisl, P. Greiner, and B. Kabashi, "Sentiklue: Updating a polarity classifier in 48 hours," in *In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014*, Citeseer, 2014.

[26] J. Xu, Y. Zhang, Y. Wu, J. Wang, X. Dong, and H. Xu, "Citation sentiment analysis in clinical trial papers," in *AMIA annual symposium proceedings*, vol. 2015, p. 1334, American Medical Informatics Association, 2015.

[27] I. C. Kim and G. R. Thoma, "Automated classification of author's sentiments in citation using machine learning techniques: A preliminary study," in *2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1–7, IEEE, 2015.

[28] Z. Ma, J. Nam, and K. Weihe, "Improve sentiment analysis of citations with author modelling," in *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 122–127, 2016.

[29] K. Ravi, S. Setlur, V. Ravi, and V. Govindaraju, "Article citation sentiment analysis using deep learning," in *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pp. 78–85, IEEE, 2018.

[30] Y. Chen, "Convolutional neural network for sentence classification," Master's thesis, University of Waterloo, 2015.

[31] M. T. Ikram and M. T. Afzal, "Aspect based citation sentiment analysis using linguistic patterns for better comprehension of scientific knowledge," *Scientometrics*, vol. 119, no. 1, pp. 73–95, 2019.

[32] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, 2006.

[33] C. A. Sula and M. Miller, "Citations, contexts, and humanistic discourse: Toward automatic extraction and classification," *Lit. Linguistic Comput.*, vol. 29, pp. 452–464, 2014.

[34] C. Jochim and H. Schütze, "Towards a generic and flexible citation classifier based on a faceted classification scheme," in *COLING*, 2012.

[35] A. Yousif, Z. Niu, J. Chambua, and Z. Y. Khan, "Multi-task learning model based on recurrent convolutional neural networks for citation sentiment and purpose classification," *Neurocomputing*, vol. 335, pp. 195–205, 2019.

[36] H. Chen and H. Nguyen, "Fine-tuning pre-trained contextual embeddings for citation content analysis in scholarly publication," *arXiv preprint arXiv:2009.05836*, 2020.

[37] C. Dong and U. Schäfer, "Ensemble-style self-training on citation classification," in *Proceedings of 5th international joint conference on natural language processing*, pp. 623–631, 2011.

[38] D. Mercier, S. T. R. Rizvi, V. Rajashekar, A. Dengel, and S. Ahmed, "Impactcite: An xlnet-based solution enabling qualitative citation impact analysis utilizing sentiment and intent.," in *ICAART (2)*, pp. 159–168, 2021.

[39] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[40] M. Honnibal and I. Montani, "Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *Unpublished software application. https://spacy. io*, 2017.

[41] H. Zou, X. Tang, B. Xie, and B. Liu, "Sentiment classification using machine learning techniques with syntax features," in *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 175–179, IEEE, 2015.

[42] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pp. 347–354, 2005.

[43] J. Wiebe and E. Riloff, "Creating subjective and objective sentence classifiers from unannotated texts," in *International conference on intelligent text processing and computational linguistics*, pp. 486–497, Springer, 2005.

[44] L. Breiman, "Random forests machine learning, vol. 45," 2001.

[45] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The weka data mining software: An update. acm sigkdd explorations news letter, 11 (1), 10-18," 2009.

[46] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella, "The 'k'in k-fold cross validation," in *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pp. 441–446, i6doc. com publ, 2012.

[47] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *the Journal of machine Learning research*, vol. 9, pp. 1871–1874, 2008.