

A COHERENT KNOWLEDGE-DRIVEN DEEP  
LEARNING MODEL FOR IDIOMATIC- AWARE  
SENTIMENT ANALYSIS OF UNSTRUCTURED  
TEXT USING BERT TRANSFORMER

BASHAR M A TAHAYNA

DOCTOR OF PHILOSOPHY  
(COMPUTER SCIENCE)

FACULTY OF INFORMATION AND  
COMMUNICATION TECHNOLOGY  
UNIVERSITI TUNKU ABDUL RAHMAN  
March 2023

**A COHERENT KNOWLEDGE-DRIVEN DEEP LEARNING MODEL  
FOR IDIOMATIC- AWARE SENTIMENT ANALYSIS OF  
UNSTRUCTURED TEXT USING BERT TRANSFORMER**

By

**BASHAR M A TAHAYNA**

A dissertation submitted to the  
Faculty of Communication and Information Technology  
Universiti Tunku Abdul Rahman,  
in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy in Computer Science  
March 2023

## **DEDICATION**

To my parents, my wife, and my lovely kids.

## ABSTRACT

### **A COHERENT KNOWLEDGE-DRIVEN DEEP LEARNING MODEL FOR IDIOMATIC- AWARE SENTIMENT ANALYSIS OF UNSTRUCTURED TEXT USING BERT TRANSFORMER**

BASHAR TAHAYNA

People can express their feelings and views via online social media like Twitter. Many fields may benefit from recognizing and evaluating the sentiments portrayed in social media content, including businesses, governments, public health, social welfare, etc. Sentiment analysis, also known as opinion mining, is a task that tries to automatically extract and classify sentiments conveyed in written content. However, this task is not always trivial especially if the written text is ambiguous or includes figurative language that deviates the meaning of the words beyond their literal meaning rather to convey a complicated meaning. Idioms are important in every natural language and people tend to use them as a shorthand to express themselves neatly. An idiom or an idiomatic expression is a set or near-set sequence of two or more co-occurring but non-contiguous words with a unified meaning or purpose. Idiomatic expressions may have literal and metaphorical meanings and are customarily known in their usual context by native language speakers. However, the literal meaning of the words that constitute the idioms often cannot be used to infer their overall purpose. The research in this thesis is motivated by the fact that idioms are underutilized in sentiment analysis, even though they typically reflect an expressive sentiment about an object or an

event. Sentiment analysis algorithms used to classify the sentiment of tweets on social media platforms such as Twitter face challenges when dealing with idiomatic expressions and figurative language used by users. These expressions often deviate from the typical meaning and sequence of words, making it difficult for sentiment classifiers to accurately classify the sentiment of a tweet. Existing methods rely on manually generated sentiment lexicons for idiomatic expressions, which requires painstaking labeling of large quantities of data, limiting their scalability and accuracy. Machine learning and deep neural networks have shown promise in accurately representing and classifying sentiment, but they require large amounts of labeled data to train the models. In this context, the proposed novel strategy aims to eliminate the need for human labeling of the idiomatic lexicon and fine-tuning the classifier to handle the sentiment classification of tweets containing idiomatic expressions. We hypothesized that revealing the implicit meaning of an idiom and using it as a feature may improve the sentiment classification results. Therefore, we proposed an idiom expansion and tweet enrichment method to integrate idioms as features in two tasks: the automatic annotation of an idiomatic lexicon and the sentiment classification of tweet data that contains idioms within it.

To evaluate the effectiveness of including idioms as features in sentiment analysis, we utilized advanced deep transfer learning techniques, including variants of the BERT (Bidirectional Encoder Representations from Transformers) model. By doing so, we sought to investigate to what extent the incorporation of idioms as features could improve the results of conventional sentiment analysis.

To begin, we selected and compiled a list of idiomatic expressions that may be assigned to a certain sentiment. Traditionally, crowdsourcing is used to manually annotate the idioms to build the gold standard sentiment lexicon of idiomatic expressions. With the promising results from our preliminary experiment, the key constraint was the substantial knowledge-engineering cost of manually creating the sentiment lexicon of idiomatic expressions which was utilized to provide idiom-based features. Therefore, we automated the development of such resources at scale to alleviate the lag time and the cost normally associated with their procurement.

The study compared the accuracy of the sentiment lexicon that was automatically annotated with the manually annotated lexicon, achieving a precision rate of 90%. The researchers then collected a dataset of tweets that included idioms and manually labeled them with a sentiment polarity to serve as a benchmark dataset. The study found that enriching the tweets with the explicit meaning of idioms led to an approximately 35% increase in classification accuracy in the sentiment analysis of the tweets dataset.

## **ACKNOWLEDGEMENT**

This thesis would not have been possible to complete without the support and guidance of the persons who helped and encouraged me in completing this research journey. I would like to present my deepest gratitude to all those people. First and foremost, I am heartfelt and truly thankful to ALLAH Almighty for His blessings and favors to make me able to achieve the objectives and remove all the obstacles in this research journey. I would like to express my sincere gratitude to my supervisor Dr. Ramesh Kumar Ayyasamy for his valuable guidance, suggestions, and support. I am truly thankful for his encouragement which makes me able to complete this research. I am heartily thankful to my co-supervisors Dr. Nur Syadhila Binti Che Lah and Dr. Rehan Akbar for their valuable support and guidance. Besides, I would also like to thank my parents, my wife, and my sons and daughters for their support and prayers in the successful completion of this research. I am also thankful to all my friends for their love, support, and concern throughout these years. In the last, I would like to present my gratitude and appreciation to UTAR for giving me the opportunity and facilities to carry out this research.

## APPROVAL SHEET

This dissertation/thesis entitled “**A COHERENT KNOWLEDGE-DRIVEN DEEP LEARNING MODEL FOR IDIOMATIC- AWARE SENTIMENT ANALYSIS OF UNSTRUCTURED TEXT USING BERT TRANSFORMER**” was prepared by BASHAR M A TAHAYNA and submitted as partial fulfillment of the requirements for the degree Doctor of Philosophy in Computer Science at Universiti Tunku Abdul Rahman.

Approved by:



\_\_\_\_\_  
Dr. Ramesh Kumar Ayyasamy  
Main Supervisor  
Faculty of Information and Communication Technology  
Universiti Tunku Abdul Rahman

Date: 17th March 2023



\_\_\_\_\_  
Dr. Nur Syadhila Binti Che Lah  
Co-supervisor  
Faculty of Information and Communication Technology  
Universiti Tunku Abdul Rahman

Date: 17/03/23



\_\_\_\_\_  
Dr. Rehan Akbar  
Co-supervisor  
Department of Computer and Information Sciences  
Universiti Teknologi PETRONAS

Date: 17/03/2023



**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 17 March 2023

**SUBMISSION DISSERTATION**

It is hereby certified that **BASHAR M A TAHAYNA** (ID No: **19ACD06906**) has completed this dissertation entitled “A COHERENT KNOWLEDGE-DRIVEN DEEP LEARNING MODEL FOR IDIOMATIC-AWARE SENTIMENT ANALYSIS OF UNSTRUCTURED TEXT DATA FROM TWITTER” under the supervision of **Dr Ramesh Kumar Ayyasamy** (Supervisor) from the Department of Department of Information Systems, Faculty of Information and Communication Technology, and **Dr Nur Syadhila Binti Che Lah** (Co-Supervisor)\* from the Department of Information Systems, Faculty of Information and Communication Technology, and **Dr Rehan Akbar** (Co-Supervisor)\* from the Department of Computer and Information Sciences Universiti Teknologi PETRONAS.

I understand that University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,




---

*(Bashar M A Tahayna)*

## DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Signature: 

Name: BASHAR M A TAHAYNA

Date: 17-March-2023

## TABLE OF CONTENTS

DEDICATION.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	vi
APPROVAL SHEET.....	vii
DECLARATION.....	ix
TABLE OF CONTENTS.....	x
LIST OF PUBLICATIONS.....	xiii
LIST OF FIGURES.....	xiv
LIST OF TABLES.....	xv
LIST OF ABBREVIATIONS.....	xvi
<b>1 INTRODUCTION.....</b>	<b>1</b>
1.1 Overview.....	1
1.2 Background.....	3
1.3 Research Problem.....	4
1.4 Research Questions.....	6
1.5 Research Objectives.....	7
1.6 Research Accomplishments and Contributions.....	8
1.7 Research Methodology.....	10
1.8 Summary.....	12
1.9 Thesis Structure.....	13
<b>2 RESEARCH BACKGROUND.....</b>	<b>15</b>
2.1 Overview.....	15
2.2 Characteristics of Textual Social Media Data.....	16
2.3 The Problem of Text Classification.....	17
2.3.1 Supervised Classification.....	18
2.3.2 Unsupervised Classification.....	22
2.3.3 Deep Learning-based Classification.....	25
2.3.4 Classification using Pretrained Embedding.....	37
2.3.5 An Overview of Key NLP Models.....	40
2.4 Data Augmentation Methods for Classification Problems.....	41
2.4.1 Paraphrasing-based approaches.....	43
2.4.2 Noising-based approaches.....	44
2.4.3 Sampling-based approaches.....	45
2.5 The Sentiment Analysis Task.....	45
2.5.1 Sentiment Analysis Levels.....	47
2.5.2 Sentiment Classification Methods.....	48
2.5.3 Sentiments of Idiomatic Expressions.....	51
2.6 Summary.....	53
<b>3 RELATED WORK.....</b>	<b>55</b>

3.1	Sentiment Classification using Machine Learning.....	56
3.2	Feature Selection and Representation.....	60
3.3	Word Embedding.....	63
3.4	Lexicon-based sentiment analysis.....	64
3.5	Approaches to building sentiment lexicons.....	69
3.5.1	Dictionary-based lexicons.....	69
3.5.2	Corpus-based lexicons.....	70
3.5.3	Human-based computing lexicons.....	72
3.5.4	Idiomatic lexicon-based sentiment analysis.....	73
3.6	Deep Transfer Learning.....	78
3.6.1	BERT Transformer.....	80
3.6.2	Transformer Fine-tuning.....	86
3.7	Data Augmentation for Sentiment Analysis Task.....	87
3.8	Summary.....	90
4	RESEARCH METHODOLOGY.....	91
4.1	Introduction.....	92
1.	Data Collection:.....	92
2.	Creation and Compilation of Idioms List Using External Online Resources:.....	92
3.	Crowdsourcing Service to Manually Annotate Idioms with Their Sentiment Polarity:.....	92
4.	Compilation of a Gold Standard Lexicon by Merging the Produced Lexicon with the SLiDE Lexicon Available at IBM Website:.....	93
5.	Extraction of Opinionated Tweets for Each Idiomatic Expression in the Lexicon Using Twitter Developer API:.....	93
6.	Expansion of Idioms by Crawling Online Thesaurus and Dictionaries to Retrieve Their Formal Definitions:.....	93
7.	Selection and Fine-tuning the BERT-variant Transformer:.....	93
8.	Identification of the Polarity Expressed in the Tweets/Idiomatic Expressions Using the Transformer, and Classify Them into Tweets as Positive, Negative and Neutral:.....	94
9.	Evaluation and Comparison of the Results of the Expansion-Based Classification with the Gold Standard Counterparts:.....	94
4.2	Methods Selection Criteria.....	96
4.3	Datasets Preparation.....	100
4.3.1	Idiomatic Expressions Preparation.....	100
4.3.2	Tweet Data Collection.....	101
4.4	Data Annotation.....	103
4.4.1	Idiomatic Lexicon Annotation.....	103
4.4.2	Gold Standard Tweets Annotation.....	106
4.5	Data Augmentation Methods.....	107
4.6	Data Expansion.....	112
4.7	Data Pre-Processing.....	115
4.8	Evaluation Measures.....	117

4.9	Summary .....	120
<b>5</b>	<b>EXPERIMENTAL RESULTS.....</b>	<b>122</b>
5.1	Experiment I: Automated Idiom Annotation Process.....	122
5.1.1	Idiom Augmentation and Expansion .....	124
5.2	Experiment II: Sentiment Classification of Tweets.....	134
5.2.1	Sentiment Classification of Tweets Dataset by Idioms Labels .....	134
5.2.2	Sentiment Classification of Tweets Dataset using roBERTa .....	135
5.2.3	Sentiment Classification Accuracy among Annotators.....	136
5.3	Experiment III: Classification Comparison of Different Deep Learning Methods.....	138
5.4	Experiment IV: Handling the Confusing Idioms .....	143
5.5	Experiment V: Performance Comparison of Lexicon Annotation using Roberta and Off-Self Sentiment Classification Tools .....	144
<b>6</b>	<b>CONCLUSION &amp; FUTURE WORK .....</b>	<b>146</b>
6.1	Overview .....	146
6.2	Summary of Results .....	146
6.3	Research Implication .....	148
6.3.1	Theoretical Implications:.....	148
6.3.2	Empirical Implications: .....	149
6.4	Future Work .....	150

## LIST OF PUBLICATIONS

Some of the work introduced in this thesis is based on the following publications:

- Tahayna, B., Ayyasamy, R., & Akbar, R. (2022). Context-Aware Sentiment Analysis using Tweet Expansion Method. *Journal of ICT Research and Applications*, 16(2), 138-151.
- Tahayna, B., Ayyasamy, R. K., & Akbar, R. (2022). ‘Automatic Sentiment Annotation of Idiomatic Expressions for Sentiment Analysis Task.’ *IEEE Access*.
- Tahayna, B. & Ayyasamy, R. K. (2020). ‘Applying English Idiomatic Expressions to Classify Deep Sentiments in COVID 19 Tweets.’ *COMPUTER SYSTEMS SCIENCE AND ENGINEERING*, 35(5), 311-319.
- Tahayna, B., Ayyasamy, R. K., Jalil, N. B. A., Sangodiah, A., Tahayna, L. N., & Krisnan, S. (2022, September). ‘Disparity-aware Pandemic Response Classification by Fine-Tuning Transfer Learning Approach.’ In *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)* (pp. 25-28). IEEE.
- Tahayna, B., Ayyasamy, R. K., Akbar, R., Subri, N. F. B., & Sangodiah, A. (2022, September). ‘Lexicon-based Non-Compositional Multiword Augmentation Enriching Tweet Sentiment Analysis.’ In *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)* (pp. 19-24). IEEE.

## LIST OF FIGURES

<b>Figure 1.1: The Generic Research Methodology</b> .....	11
<b>Figure 2.1: An Example of a Fully Connected Feed-Forward Network</b> .....	29
<b>Figure 2.2: RNN Network Layers</b> .....	31
<b>Figure 2.3: Neurons Linking in the Convolutional Neural Network</b> .....	33
<b>Figure 2.4: CNN Architectural Design</b> .....	33
<b>Figure 3.1: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings, and the position embeddings (Source: BERT’s paper, Devlin et al., 2018)</b> .....	83
<b>Figure 3.2: Overall pre-training and fine-tuning procedures for bi-directional encoder representations from the transformer (BERT) (Source: BERT’s paper, Devlin et al., 2018)</b> .....	87
<b>Figure 4.1: The Proposed Framework</b> .....	99
<b>Figure 4.2: Sample of Retrieved Tweets using Idiomatic Expressions in the API Query</b> .....	102
<b>Figure 4.3: Annotation platform interface</b> .....	106
<b>Figure 4.4: Sentmixup Approach Combines Existing Samples by Averaging Sentence Embeddings</b> .....	112
<b>Figure 4.5: Confusion Matrix for Binary Classification</b> .....	118
<b>Figure 5.1: The Semantic Relatedness of Terms in the Word2Vec Embedding Space</b> ..	126
<b>Figure 5.2: The Output of the MLM Task using BERT</b> .....	127
<b>Figure 5.3: Transformer-Based Back-Translations (English-Hindi-English)</b> .....	128
<b>Figure 5.4: Transformer-Based Back-Translations (English-German-English)</b> .....	129
<b>Figure 5.7: Accuracy of BERT, Wiki2Vec, and ELMo Embedding Methods as the Number of Epochs Increases (enriched tweets)</b> .....	140
<b>Figure 5.8: Accuracy of the Hybrid CNN-LSTM Model Using the ELMo Embedding</b> ..	141
<b>Figure 5.9: Comparison of the F-Score Results of the CNN-LSTM Hybrid Model, CNN, and LSTM Concerning Different Embedding Techniques</b> .....	143

## LIST OF TABLES

<b>Table 2.1: Pros &amp; Cons of Lexicon-Based Classification Methods</b> .....	25
<b>Table 2.2: One-Hot Encoding Example</b> .....	28
<b>Table 2.3: Documents Representations using Bag-of-Words</b> .....	28
<b>Table 2.4: Summary of the Paraphrasing-based Data Augmentation</b> .....	43
<b>Table 3.1: Sample word list from the MPQA lexicon</b> .....	66
<b>Table 3.2: Sample Keyword Annotations in VADER Lexicon</b> .....	67
<b>Table 3.3: Parameters Setting of BERT-Base and BERT-Large</b> .....	86
<b>Table 4.1: Sample of the collected idioms from online dictionaries and thesauruses</b> ....	100
<b>Table 4.2: Pseudo Code of Tweet Collection Algorithm</b> .....	102
<b>Table 4.3: eSLiDE Idioms’ Polarity Distribution</b> .....	103
<b>Table 4.4: The SLiDE Lexicon Columns Structure</b> .....	104
<b>Table 4.5: Sample Content of the SLiDE Lexicon</b> .....	107
<b>Table 4.6: Pseudo Code of the Idiom Expansion Procedure</b> .....	114
<b>Table 5.1: Sentiments’ Polarity Distribution of Ground Truth Tweets Dataset</b> .....	122
<b>Table 5.2: Sample of Idiom Expansion from Two Different Resources</b> .....	131
<b>Table 5.3: Sample of eSLiDE Lexicon Automatic Annotation of Idioms Using Twitter-Roberta-Base-Sentiment Classifier without Expansion</b> .....	131
<b>Table 5.4: Sample of Twitter-Roberta-Base-Sentiment Sentiment Classification With/Out Idioms Expansion</b> .....	133
<b>Table 5.5: Comparison of Error Ratio While Annotating Idioms Without-Out Idiom Expansion</b> .....	133
<b>Table 5.6: Precision, Recall, and F1-Score Results Using roBERTa</b> .....	133
<b>Table 5.7: Annotation of the Sentiment Lexicon of Idiomatic Expressions Using the Twitter-Roberta-Base-Sentiment Transformer and Different Augmentation Methods</b> 134	134
<b>Table 5.8: Error Percentage of Direct Sentiment Assignment Based on Idiom Label</b> ... 135	135
<b>Table 5.9: Accuracy of Tweet Sentiment Classification Using the Twitter-Roberta-Base-Sentiment</b> .....	136
<b>Table 5.10: Accuracy and Consistency of the Tweets Classification Using the Automatic-Based Annotation Method</b> .....	138
<b>Table 5.11: Precision, Recall, and F-score Comparison of Raw and Enriched Tweets</b> . 142	142
<b>Table 5.12: F1-Sore Comparison of the Expansion Method Using Single and Multi-Definition Methods</b> .....	144
<b>Table 5.13: Sample of the Sentiment Lexicon Annotation before Expansion Using Different off-shelf Sentiment Classifiers</b> .....	145
<b>Table 5.14: Idiom Annotation Comparison: roBERTa vs. CoreNLP, and SentiStrength Tools</b> .....	145



## LIST OF ABBREVIATIONS

<b>SA</b>	Sentiment Analysis
<b>MLM</b>	masked language model
<b>NSP</b>	Next sentence prediction”
<b>NLP</b>	Natural Language Processing
<b>BERT</b>	Bidirectional Encoder Representations from Transformers
<b>UGC</b>	User Generated Content
<b>CNN</b>	Convolutional Neural Networks
<b>RNN</b>	Recurrent Neural Network
<b>TP</b>	True Positive
<b>TN</b>	True Negative
<b>FP</b>	False Positive
<b>FN</b>	False Negative
<b>API</b>	Application Programming Interface
<b>LLRD</b>	Layer-wise Learning Rate Decay
<b>SWA</b>	Stochastic weight averaging
<b>CSV</b>	Comma-Separated Value
<b>CBOW</b>	Continuous Bag-of-Words
<b>VADER</b>	Valence Aware Dictionary and Sentiment Reasoner

## LIST OF EQUATIONS

<b>Equation 4.1</b>	Krippendorff's alpha coefficient.....	97
<b>Equation 4.2</b>	Precision and Recall Calculations.	110
<b>Equation 4.3</b>	F-Measure metric equation.....	110
<b>Equation 4.4</b>	Accuracy metric calculation.....	110
<b>Equation 4.5</b>	Error ratio.....	111
<b>Equation 5.1</b>	Error rate equation.....	122

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

The computational analysis of user-generated content (UGC) on Web 2.0, such as the textual forms of social media (e.g., status updates, and postings), offers an alternative or supplementary method to conventional qualitative research techniques such as questionnaires, interviews, and structured observations. It enables a large variety of practical applications to collect and analyze UGC to comprehend people's or customers' perspectives on any topic, product, or service (e.g., social events, market research). The tremendous quantity of textual content that we can obtain via the use of the internet makes the difficulties of qualitative analysis much more difficult. Text mining has emerged as a feasible remedy to the issue of information overload created by enormous volumes of text derived from a variety of sources. More recently, opinion mining, which is another name for sentiment analysis, appears as an effective solution to mitigate the analysis problem that has been caused by an abundance of text. Sentiment analysis is a process that can automatically extract and classify views and perspectives that are expressed in a piece of text.

In their research paper, Strapparava and Mihalcea remark that any lexeme or other lexical units can provide vital insight into the intended meaning of communicated messages [1]. The difficulty in analyzing large volumes of textual content, which is commonly found on social media platforms and other web 2.0 sources. Traditional qualitative research techniques, such as

questionnaires and interviews, may not be sufficient to handle such a vast amount of information. This difficulty in analysis is addressed through the use of automated approaches, such as sentiment analysis.

Sentiment analysis, also known as opinion mining, is a process that can automatically extract and classify views and perspectives expressed in a piece of text. By using specialized lexicons, such as WordNet-Affect [2], sentiment analysis can recognize affective notions that transmit feelings or emotions. The approach employs a wide range of linguistic elements and features, such as words, part-of-speech tagging, morphological and lexical analysis, and grammatical relationships.

The automated approach of sentiment analysis provides a feasible remedy to the issue of information overload created by enormous volumes of text derived from various sources. Business organizations can utilize sentiment analysis to extract important information from user-generated content on social networks to expedite their expansion, while customers and organizations can gain a deeper understanding of client thoughts or attitudes.

In summary, sentiment analysis offers an alternative or supplementary method to conventional qualitative research techniques, enabling the collection and analysis of user-generated content on web 2.0 platforms. By providing a way to automatically extract and classify views and perspectives expressed in large volumes of text, sentiment analysis provides an effective solution to the problem of information overload created by enormous volumes of text [3].

## 1.2 Background

Even though “more than 10% of the data obtained from Twitter is comprised of idiomatic expressions” [73], sentiment analysis has not widely highlighted the idiomatic expressions as a significant feature for tweet data. Idioms reflect a semantic orientation and a sentiment polarity toward an object or an event [4, 5]. The idiom “To be in good humor” means a genial disposition or mood. This means an idiom may directly reveal a scene's mood. However, idioms usually preserve their meaning as a single semantic unit [4, 6, 7], and therefore, when using idioms directly to reveal a sentiment, the classification error generally indicates that utilizing idioms' keywords misleads the classifier. For example, the “B's 'n' E's” idiom refers to something or someone excellent. This means that emotion and sentiment can't always be revealed by the constituting words. To handle these cases, some researchers propose a lexicon-based sentiment analysis by manually building sentiment lexicons of idiomatic expressions using a crowdsourcing service. However, this is a time and resource-consuming process. Therefore, much of the existing research is restricted to assessing the literal meaning of idioms and ignores the non-literal implicit meaning of colloquial phrases.

The concrete motivation behind this thesis is to make this research a significant contribution to the literature by proposing “an automatic creation and annotation of a **sentiment lexicon of idiomatic expressions.**” Our motivation is that idiom-based features can improve sentiment analysis, and idioms' sentimental polarity can be derived from their meaning or definition. We automated the feature engineering process by proposing an idiom-expansion method to reduce the bottleneck associated with lexico-semantic resource

acquisition. In addition, recent research trends on “pre-trained transformer”-based sentiment analysis stress the need for fine-tuning to carry out the task. However, the viability of fine-tuning stability is still called into question by critics. According to the findings of some research [8], the instability of the fine-tuning process can be attributed, in part, to problems with vanishing gradients and a lack of generality. Other researchers have pointed the finger of blame at the catastrophic forgetting nature of the transformers and the limited scope of the datasets [9, 10, 11], calling into doubt if these are the core causes of the instability. Therefore, we propose a tweet-enrichment strategy to eliminate the re-training or fine-tuning of the transformer by enriching the context of a tweet with a self-explanatory intention or purpose of the idiom contained within a tweet text.

### **1.3 Research Problem**

On Twitter, users tweet their opinions about various topics using natural language. Frequently, they express themselves using informal writing styles and employ shorthand or figurative language. This includes the usage of phonetically similar words, acronyms, or abbreviations. Figurative language refers to the practice of using words in a manner that deviates from their typical sequence and meaning to convey a more refined feeling, vivid writing, greater clarity, or emotional contrast. Figurative language such as idioms, often known as idiomatic expressions, have seen a surge in popularity on social media platforms such as Twitter. Using idioms, users can refer to something without directly stating what it is by using ordinary common terms or phrases [12].

Sentiment analysis algorithms may be used to computationally classify and discern the sentiment of a tweet. The sentiment classifier, on the other hand,

must be aware of the nonliteral meaning of idioms and deal with the continually growing and evolving meaning and use of idioms or slang. Previous scholars manually develop and use lexicon for sentiment analysis to classify idiomatic expression sentiments. To obtain a high degree of classification accuracy, large-scale and high-quality sentiment lexicons are necessary. However, the manual generation and maintenance of a sentiment lexicon for idiomatic expressions requires painstaking labeling of a large quantity of data. Therefore, little research on sentiment analysis that employs lexicons has been conducted.

Methods based on machine learning or deep neural networks extract and represent information, which may aid in more accurate sentiment classification. Traditional machine learning requires human labeling and compilation of “good enough” training data to represent knowledge. On the other hand, deep learning needs a massive quantity of data to train the model. Pre-trained transformers are an excellent option for avoiding model training from scratch. Transformers, contain millions of parameters and are often trained to tackle particular tasks. To handle a new downstream task, retraining and fine-tuning the transformer are commonly required procedures. Unfortunately, even fine-tuned retraining transformers may fail to appropriately classify the sentiment of idiomatic expressions if they are unaware of the concealed meanings and contextual usages of the updated idioms. To address the aforementioned issues, we propose a novel strategy that eliminates the necessity for human labeling of the idiomatic lexicon required to retrain a transformer by eliminating the need to fine-tune the classifier to handle the sentiment classification of tweets including idiomatic expressions.

## 1.4 Research Questions

In this work, we aim to enhance sentiment analysis of tweets containing idiomatic expressions. To support our research, we leverage external knowledge bases to acquire relevant and significant information about idioms and enrich the context of tweets. External knowledge bases refer to any sources of information that are not explicitly contained within the data being analyzed, but that can provide additional context, insights, or knowledge that can be used to enhance the analysis [15]. In the context of sentiment analysis of tweets containing idiomatic expressions, external knowledge bases could include a variety of resources, such as online dictionaries, thesauruses, or online idiomatic lexicons [13].

Online dictionaries can provide definitions and explanations of individual words, including their literal and figurative meanings, as well as their connotations and nuances. Thesauruses can provide synonyms and antonyms for specific words or phrases, which can be useful in identifying related concepts and sentiment associations. Idiomatic lexicons can provide collections of idiomatic expressions, along with their definitions and examples of usage, which can help to identify sentiment associations and understand the nuanced meanings of idiomatic expressions.

By leveraging external knowledge bases, sentiment analysis can benefit from a broader and more nuanced understanding of the language used in tweets containing idiomatic expressions. This can help to identify sentiment



associations and accurately classify sentiment, even in cases where the meaning of the words is ambiguous or non-literal. To achieve this goal, we investigate the following research questions, which are further supported by relevant literature:

**RQ1:** How can we efficiently build and annotate a sentiment lexicon of idiomatic expressions using external knowledge bases?

**RQ2:** What is the impact of incorporating idiomatic expressions as features on the sentiment classification of tweets?

**RQ3:** How does leveraging external knowledge bases enhance the performance of sentiment analysis of tweets containing idiomatic expressions?

**RQ4:** How to perform a sentiment classification of tweets with idiomatic expressions while having little or no training data?

**RQ5:** To what extent does the use of data augmentation and normalization pre-processing procedures influence the accuracy of the sentiment classifier?

## **1.5 Research Objectives**

In this thesis, we develop a method to automatically analyze textual content shared on Twitter and classify people's opinions or sentiments into positive, neutral, or negative polarities. To be precise, we developed a novel framework for sentiment classification of tweets containing idiomatic expressions. This framework could make it possible to get a profound insight and understanding of what individuals believe or feel by classifying the sentiment represented in their tweets. In light of this, we present the following list of objectives:

**RO1:** Develop an automated method for building and annotating a sentiment lexicon of idiomatic expressions to eliminate the need for manual annotation.

**RO2:** Investigate and compare different methods for incorporating idiomatic expressions as features in sentiment classification of tweets to determine the most effective approach.

**RO3:** Evaluate the impact of leveraging external knowledge bases on the performance of sentiment analysis of tweets containing idiomatic expressions.

**RO4:** Develop and test methods for sentiment classification of tweets with idiomatic expressions in situations with limited training data.

**RO5:** Investigate the influence of data augmentation and normalization pre-processing procedures on the accuracy of the sentiment classifier.

## **1.6 Research Accomplishments and Contributions**

The study's contribution is the automation of the creation and annotation of a sentiment lexicon of idiomatic expressions, which has not been extensively explored in sentiment analysis. The study uses an expansion method for idioms and shows that it enhances sentiment classification outcomes substantially.

The study also provides a large collection of nearly 4,000 idioms that have been carefully annotated with sentiment polarity and have a reliable inter-annotation agreement, which is one of the biggest sentiment lexicons of idiomatic expressions of its sort that can be used for sentiment analysis tasks.

Additionally, the study proposes a technique to computationally extract sentiment from dictionary definitions of idioms to automate the acquisition of

their sentiment polarity, which can be used to expand the lexicon and make it possible for it to be ported to other languages.

Overall, the novelty or new knowledge creation of this study lies in its contribution to sentiment analysis by exploring the role of idioms and providing a technique for the automation of sentiment lexicon creation and annotation. It fills the gap in the body of knowledge by expanding the scope of sentiment analysis and providing a resource for researchers in the NLP community to further explore and exploit idioms as features of sentiment analysis. This research gap has been closed by our work, which also makes a substantial contribution to the existing body of knowledge which can be summarized into:

- a. The first contribution that may be made as a result of this study is the automation of the annotation process of the sentiment lexicon of idiomatic expressions. The proposed idiom expansion method can be applied to other languages as it does not require specialized toolkits to pre-process the idiomatic expressions. The recent research focuses attention on the manual creation of sentiment lexicons to be used directly as a reference in the lexicon-based sentiment analysis or to be used for training machine learning methods. The manual process is tedious and time-consuming.
- b. The thesis compares different off-shelf sentiment analysis tools with transformer-based classification and highlights the benefits of the idiom expansion method for achieving higher classification performance.
- c. Third, the research that is being presented here identifies the

importance of idiomatic expressions in determining the overall sentiment of tweets containing them.

- d. The fourth contribution is that the study suggests the avoidance strategy of retraining and fine-tuning the pre-trained transformer by employing tweet enrichment, using the idiomatic lexicon or even on a real-time basis, to enhance the sentiment classification accuracy.
- e. Finally, the study reports the actual impact of the previous common pre-processing and data augmentation methods on the classifier performance.

## **1.7 Research Methodology**

The purpose of this research is to acquire a comprehensive grasp of the fundamentals, methodologies, and obstacles involved in enhancing sentiment classification using idiomatic expressions as features by conducting a review of different methods used for sentiment analysis. This helps to develop a comprehensive and holistic perspective of the numerous sentiment analysis areas, which makes the deployment of desired processes easier in later stages. Figure 1.1 provides a generic overview of the study approach, which may be broken down into two stages, each of which has multiple steps. These stages are examined in further depth in the next chapters. In the first step, we will focus on defining different characteristics of the dataset and then using those definitions to annotate the dataset. The following are the steps that comprise this phase, which can be summarized as follows:

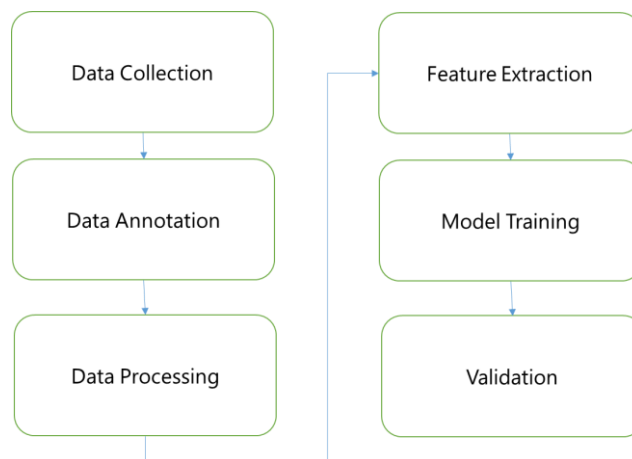
1. Determine the idiomatic expressions to be used.
2. Gather the tweets for the chosen target idioms.

3. Analyze and filter tweets that are relevant to this study.
4. Use the “idiom expansion method” to annotate idiomatic expressions to retrieve their definition (s). This includes using external knowledge bases to automatically assign sentiment scores to extracted opinion words.
5. Validate the annotated tweets and assigned sentiment polarities to generate the dataset and idiomatic sentiment lexicons.

The second step comprises the development of an integrated idiomatic lexicon and deep learning model for the classification of sentiments or opinions.

The steps comprising this phase can be stated as follows:

- a. Feature extraction and representation for the source tweet.
- b. Utilize the deep learning model for training/testing/validation.
- c. Assign the results of the scoring sentiment.
- d. Aggregate, summarize, and present the classification outcomes.



**Figure 1.1: The Generic Research Methodology**

## 1.8 Summary

The rise of social media has resulted in a deluge of user-generated data that can be mined for insights into public sentiment and the thoughts and experiences of billions of internet users. The purpose of sentiment analysis is to identify the positive, negative, and neutral feelings about a topic expressed in a piece of writing. It finds use in business settings such as marketing, public relations, and customer service. The present state of the art in sentiment analysis may be divided into two categories: those that use machine learning and data mining to train a model using a collection of labeled data (supervised learning), and those that do not (unsupervised learning). In contrast, lexicon-based systems detect attitudes by comparing text terms with prepared lexicons, assigning each word a weight depending on the polarity to which it belongs.

This research makes use of the sentiment lexicon of idiomatic expressions together with a deep learning transformer-based sentiment analysis, focusing on the idiomatic expressions found in five of the most Twitter data. The proposed approach is different from the traditional usage of the lexicon-based method in several ways: 1) the sentiment lexicon acts as a swift bridge to enrich a tweet context by importing the expanded form of idiomatic expression in the lexicon. This means that our idiom expansion method can work in real-time and enrich tweets directly if we employ the idiom detection method. 2) Unlike existing prominent lexicons in the area of sentiment analysis such as (SliDE, VADER, SentiStrength, SentiWordNet, Liu and Hu opinion lexicon, and AFINN-111), our lexicon was annotated automatically without any human intervention.

Therefore, the research problem is used as a basis for developing the research questions as well as the research objectives. The major reason for

conducting and completing this research is to propose a strategy for automatically creating and annotating a sentiment lexicon and using it to improve the classification of sentiments of tweet data containing idiomatic expressions. By comparing the classification error ratio or the accuracy, and the F1-measure, we evaluate the efficacy of different lexicons in Twitter polarity classification. Neutral, positive, and negative sentiments were classified with more precision using our proposed method.

## 1.9 Thesis Structure

The overview of the remaining thesis organization is presented as follows:

***Chapter 2: Research Background*** – The fundamental concepts, ideas, and principles behind sentiment analysis are presented in this chapter.

***Chapter 3: Review of Literature*** – The existing literature on sentiment analysis with the utilization of idiomatic expressions has been discussed. Moreover, the existing state of knowledge and limitations of the existing literature has been highlighted in this chapter.

***Chapter 4: Proposed Framework*** – The steps and tasks involved in building the sentiment analysis framework have been outlined. This chapter explains the primary parts, subparts, and modules to create and build the sentiment of the idiomatic expressions lexicon and the expansion modules that have been utilized to enhance the sentiment classification task by enriching tweets' contexts.

***Chapter 5: Experimental Results & Discussion*** - Results In this chapter we present the classification results and the gained enhancement to verify the proposed model. The outcomes have also been examined in light of the research aims and objectives. In this chapter, we've shown how the suggested framework may be put into practice by providing concrete instances of the idiomatic lexicon in action.

***Chapter 6: Conclusion & Future Work*** – In this chapter, the conclusion and summary of results that are consistent with the findings of the study have been provided. Additionally, the future paths of research have been outlined for the reader.



## CHAPTER 2

### RESEARCH BACKGROUND

Sentiment analysis is defined and introduced in this Chapter. The chapter is divided into the following parts: In Section 2.1, we'll go over the basics of what sentiment analysis is and how it might be used. Section 2.2 provides an overview of the nature and features of user-generated content on social media platforms, as well as the primary reasons why people write in shorthand and use figurative language. Section 2.3 covers the text classification issue from the perspective of natural language processing, as well as the representation and classification of sentiment and their polarities. In Section 2.4, we describe the approaches of data augmentation that are often used in sentiment analysis assignments. These strategies might be used to generate textual data with alternate terms and keywords while preserving the textual data's context. Section 2.5 gives a briefing on the contemporary techniques used in sentiment analysis. It explains the levels and primary techniques used in this discipline. In addition, this part introduces the notion of the implicit sentiment contained inside idiomatic expressions, as well as the problems associated with identifying them.

#### 2.1 Overview

According to Statista, “more than 4.26 billion individuals were using social media all over the globe in 2021,” [14]. It is expected that the number would reach close to six billion by the year 2027 [14]. Social media platforms are among the most important tools of contemporary communication, and they

have evolved into a *de facto* publishing medium for businesses, organizations, and governments [15]. By offering a bidirectional communication channel between institutions and their current or prospective audiences, social media sites form an interactive communication environment between all parties. Social media platforms have resulted in a tremendous increase in the quantity of data and information that is now accessible. The proliferation of social media has been responsible for the fast growth of tasks and applications related to natural language processing (NLP), such as sentiment analysis, and has led to the creation of various brand-new applications. Sentiment analysis is a process of extracting features from textual material, accurately representing those features, and then classifying them according to a predefined set of emotional polarities or classes.

## **2.2 Characteristics of Textual Social Media Data**

Online social networking is a significant modern activity because it enables individuals and businesses to create connections that would otherwise be impossible owing to geographical or temporal distances. This proximity fosters engagement, which in turn promotes cultural understanding and also aids to enhance business productivity. On social media platforms, awareness of key components of cultures is important to comprehend the cultures of others. For example, users in the United Kingdom may express their thoughts and opinions on a subject by utilizing linguistic qualities or local language terminologies that have a distinct application or meaning for users in the United States.

The use of natural languages, and sometimes sophisticated linguistic features, to convey one's idea or intention is a common way for people to voice their opinions about products or services that they have encountered when

shopping online. However, the number of words that may be used to form a single message is restricted on some platforms, including Twitter. Tweets are frequently shortened, and because of this, it is not always easy to comprehend them computationally without their context. The shorthand method takes into consideration restrictions on the number of characters that may be used and “condenses the required message into a small number of phonetically related words or symbols” [15]. Users will thus resort to the use of abbreviations or figurative language wherever possible to be more effective, persuasive, and impactful. The use of idioms, euphemisms, and slang may provide an air of friendliness, civility, and even humor to an otherwise intense discussion. Another reason for using shorthand may be to reduce the risk of having a criminal conviction or legal action in case of the use of blatantly “inappropriate” words. Therefore, comprehending or classifying the purpose of a tweet using computational algorithms is not always an easy task [16]; tweets are comprised of several intricate facets and frequently run into issues such as: “a) the unstructured nature of the data; b) the multilingual aspects; c) incomplete sentences; d) the application of idioms, jargon, or ad hoc words; e) lexical vs. semantic vs. syntactic attributes; f) ambiguity; and the implicative references or meanings,” [17].

### **2.3 The Problem of Text Classification**

Predictive data mining is one of the most powerful applications that use machine learning. Text classification by machine learning is an example of a predictive modeling challenge. In text classification, a collection of binary, categorical, or continuous features representing each instance in the dataset may be used to train the machine, while instances of the input data are assigned

predetermined class labels. In supervised learning, the learning process needs training datasets containing many input and output instances. The notion of learning refers to how a model can optimally map input data instances to a certain output class. The training data should be adequately depicting the problem and provide instances for each class label. Classification applications include document classification by subject, email classification by spam status, and classification of handwritten characters, among others. In contrast, unsupervised learning attempts to classify unlabeled occurrences into “similar” categories using unlabeled examples. In the following subsections, we explain the different learning paradigms used in text classification.

### **2.3.1 Supervised Classification**

The purpose of developing a function using supervised learning is to predict a given label based on the data that is supplied. This may be done by categorization or predictive modeling. The regression method provides an estimate of continuous output, while the classification model provides an estimate of discrete class labels. The classification model does this by using an item's attributes to establish which category the item belongs in. Classification and regression are two statistical procedures that may be carried out using several approaches. Classification is just regression with a threshold added to it, while regression can be thought of as the foundation for classification. If the number in question is more than the criterion, then it is accepted as an accurate representation. It is regarded to be false if the value is lower than the threshold that was established. Based on the application and objective of the classification, the following are examples of the main supervised classification methods.

### 2.3.1.1 Binary Classification

The objective of a binary classification algorithm, which is to determine which of the two classes the data will fall into. The two classes could be represented by values such as 1 and 0, or true and false. Many different machine learning algorithms have the capability of finding answers to problems of this kind. Both the amount of the data (i.e., the total number of instances that are accessible) and the quality of the data (i.e., the features, imbalanced data, outliers, etc.) are critical factors in determining how well the algorithm works.

### 2.3.1.2 Multiclass Classification

In multiclass classifications, prediction is made on one and only one of more than two exclusively mutual classes. Consider the following representation for the collection of output classes of size  $n$ ,  $C = \{C_1, C_2, C_3, \dots, C_n\}$ . To make a prediction, an element  $I_x$  from the input set  $I = \{I_1, I_2, \dots, I_m\}$  is mapped into a specific class  $C_y$ , where  $y$  is the index of the correct class in the set  $C$ .

Multiclass classification examples include color classification and handwritten character classification. For example, in color classification, the goal might be to classify an input image into one of several color categories, such as red, green, blue, or yellow. Similarly, in handwritten character classification, the goal might be to classify an input image of a handwritten character into one of several possible character classes, such as letters of the alphabet or digits [201].

### 2.3.1.3 Multilabel Classification

In multi-label classification, the objective is to forecast or predict one or

more classes when one or more class labels are anticipated for each instance. In contrast to multiclass classification, each label represents a distinct classification job that is yet connected. For instance, depending on an advertising poster, the genres of a film may be classified and given labels such as “nature,” “documentary,” “horror,” and “adventure.”

#### **2.3.1.4 Imbalanced Classification**

In this context, the number of training instances belonging to each class is not distributed evenly. That is the case when the distribution of the classes is uneven, not near to equal, but considerably skewed or biased. Any method that is often used to solve problems with machine learning typically leads to incorrect outcomes. Therefore, researchers come up with a variety of proposals and approaches to solve the issue of imbalanced data. The following are some of the most common approaches:

**Choosing An Appropriate Measurement Standard:** because it reflects a harmonic mean of accuracy and recall, the F-score metric to measure the accuracy is better suited for usage with a dataset that contains classes that are not evenly distributed. The F-score maintains the equilibrium between accuracy and recall and only increases the score if the classifier correctly identifies more examples of a certain class.

**Oversampling And Undersampling:** this method is used to resample the majority class and the minority class respectively [18]. There are two approaches to implement this strategy: (1) by “adding” instances to the data to generate more data samples to the minority class (to oversample the minority

class), or (2) by deleting instances from the data in majority class (to downsample the majority class).

Searching For The Ideal Value And Thresholding: given that most classifiers tend to forecast the likelihood of class membership, the prediction is allocated to a certain class based on a threshold that is typically set to 0.5. To ensure that this method is capable of effectively dividing classes into distinct groups, we will need to adjust the point until it reaches the ideal value. The threshold of 0.5 is commonly used as a default threshold value in binary classification problems, where the objective is to predict the class membership of a binary outcome variable (e.g., yes/no or true/false). The threshold represents the point at which the classifier should assign a particular data point to one of the two classes, based on the predicted probability of class membership.

When the predicted probability of class membership is greater than or equal to the threshold (0.5), the classifier will assign the data point to the positive class; otherwise, it will assign it to the negative class. This threshold is often used because it represents a natural and intuitive point of division between the two classes.

However, in imbalanced classification problems, where the distribution of the outcome variable is highly skewed towards one class, a threshold of 0.5 may not be appropriate. In such cases, the threshold can be adjusted to achieve better classification performance.

The ideal value of the threshold will depend on the specific problem and the desired trade-off between different performance metrics such as sensitivity, specificity, precision, and recall. Therefore, searching for the ideal value of the

threshold is an important step in optimizing the performance of an imbalanced classification model.

### **2.3.2 Unsupervised Classification**

Unlike supervised classification, unsupervised classification refers to the clustering techniques used in an “unsupervised learning manner” to identify similarities between data points to reveal previously hidden connections between variables. Through the use of feature learning, in which the learning algorithm is not provided with any labels, it seeks to unearth previously hidden patterns in the data. Although there are several methods for unsupervised learning, the most relevant to the sentiment analysis context is the lexicon-based method.

#### **2.3.2.1 Lexicon-Based Classification**

Lexicon-based classification is a technique of classifying documents based on “the number of words from lexicons associated with each class label” [19]. Usually, labels are assigned to documents by comparing the frequency with which terms from two opposing lexicons, such as positive and negative sentiment or emotions, occur in the text. Creating such word lists is often simpler than labeling examples, and non-experts may troubleshoot them if classification performance is insufficient. This classification is heuristic, however, lacks analysis and rationale. The notion that all terms in any lexicon are equally predictive is essential to lexicon-based classification. This is seldom the case in reality, which is why supervised classifiers that learn separate



weights for each word from labeled instances often outperform lexicon-based techniques.

The learning algorithm begins the process of constructing the lexicon without being supervised. In the context of sentiment classification, the algorithm uses a function to evaluate the degree to which a text unit is positive by taking into account both the positive and negative signs that are located inside the unit. A few scholars have considered the idea of classifying words based solely on their synonyms without human supervision [20-23].

The application of this approach has the benefit of not requiring the collection of any data for training purposes, which is a significant advantage. The major drawback is that the technique is dependent on the domain. Certain business domains have words and jargon that other business domains use or utilize differently. [24]. In case the principal scoring method is unable to correctly classify a certain term, several fascinating alternatives might be used. For example, we may utilize the polarity of the statement that came before it as a tie-breaker [25]. Table 2.1 presents a comparison of the various strategies, highlighting both their strengths and weaknesses.

- **Corpus-based strategy:** This method analyzes the syntactic patterns and the semantic information of a phrase or a sentence to compute its sentiment. After beginning with a predetermined list of polarity and sentiment terms, this technique searches a huge corpus for syntactic or other patterns that are comparable to finding sentiment tokens and the orientation in which they are expressed. This strategy is optimized for training with a

substantial quantity of labeled data and is customized to a certain domain. Nevertheless, it is helpful with opinion words that have context-dependent orientations [25].

- **Dictionary-based strategy:** The approach based on dictionaries makes use of a collection of terminology that has been painstakingly constructed by hand and contains sets of views that have already been decided [26, 27]. The key premise behind this technique is that antonyms possess polarities that are opposite to those of their source words, but synonyms possess polarities that are similar to those of their sources. Large corpora are searched, such as a thesaurus or WordNet, to add antonyms and synonyms to a group or seed list that has already been produced [25].
- **Idiomatic-aware strategy:** This method is similar to the dictionary-based process. The difference is that the collection comprises figurative language components or expressions. In this case, the individual word is not considered in isolation- because the use of words (or synonyms/antonyms) usually deviates from the conventional meaning. In addition, such expressions hold a complicated meaning that conceals the actual sentiment different from what individual words might suggest. This method presumes that- most of the time, the idiomatic expressions hold the same implicit meaning regardless of the context they are used within.

**Table 2.1: Pros & Cons of Lexicon-Based Classification Methods**

<i>Lexicon</i>	<b>Pros</b>	<b>Cons</b>
<i>Dictionary</i> [129]	<p>The need for trained data is eliminated.</p> <p>Provide satisfactory results in areas with little data.</p> <p>Dictionary definitions at your fingertips</p>	<p>Consists of opinionated words with a focus on a particular topic</p> <p>Inability to locate opinion words from a certain domain that are not in the lexicon</p>
<i>Corpus</i> [129]	<p>The ability to recognize opinionated expressions with a certain content slant.</p> <p>Archives best results when domains are separated.</p>	<p>The wide span of the corpus causes wide variations in performance.</p> <p>Because it's impossible to provide large texts and cover all text keywords, they can't be utilized separately.</p>
<i>Idiomatic</i>	<p>Domain-independent usage of idiomatic expressions.</p>	<p>Specifically developed for idiomatic expression.</p> <p>It is necessary to have an extraction mechanism to retrieve expressions from the input text.</p>

### 2.3.3 Deep Learning-based Classification

#### 2.3.3.1 Deep Learning for NLP

The development of intelligent systems is made possible by the swift and significant advancements in hardware and software technology. Intelligent systems are highly advanced devices that can sense and respond to their surroundings. There are many different forms of intelligent systems. Examples of machines that can perceive and engage with their surroundings include smartphones, robots, and cameras. Digital cameras can detect and recognize faces, smartphones can convert voice to text, and self-driving innovative automobiles can recognize and avoid impediments in their route. Thankfully,

progress also includes several NLP applications. NLP chatbots are a popular application. Because neural networks are so powerful, the chatbot may utilize them to recognize incoming text, summarize materials, and even create new creative text.

Since the 1950s, computer scientists and engineers have been researching and experimenting with neural networks. The initial breakthrough, however, takes around thirty years. The second significant innovation occurred thirty years later, in 2012, during the so-called deep-learning revolution. The amount of data and computer power increased that year. It was also discovered and realized that neural networks with several layers, as opposed to only a few layers, might perform noticeably better [28] (LeCun, et al., 2015).

The capacity of neural networks to classify and identify input data, and more recently, the potential of certain network topologies to generate novel content, has garnered a lot of attention in recent years. These advancements were made possible via an artificial intelligence approach called deep learning. Deep learning relies on a neural network data structure roughly modeled to mimic the networks of real neurons in the human brain; similar to how neurons in the brain function, neural networks do as well [29] (Ruder, et al., 2016). In this architecture, the input from one layer is connected to an output from a subsequent layer where each neuron takes in data, analyzes it internally, and then produces a result to the next layer with the hope that the output is more closely matched to the intended outcome (in the case of labeled data).

Deeper neural networks should be more adaptive, according to theory. Deeper networks, however, would have been complex for the raw computer

power to handle. But more significantly, there was no practical method for training deep neural networks. The original neural networks' straightforward hill-climbing algorithms were not saleable for deeper networks. Numerous studies have shown that neural networks can learn complex operations in various fields. The major weakness of neural networks is that they are naturally strong; if the learning process is not carefully designed, neural networks typically easily overfit the training data. In the actual world, overfitting happens when a neural network performs well on training examples but poorly on untried test instances [35].

Within the realm of natural language processing, computers attempt to study and comprehend human language to carry out practical tasks. In this way, they might glean useful data from texts. After a series of failed attempts at using handwritten rules by deduction and abstraction reasoning (see [30]), neural networks were developed to discover these rules on their own. Neural networks and other machine learning algorithms, however, cannot process anything except quantitative input; so, we must devise a means of translating the text we want to evaluate into numbers. Numerous options exist for achieving this goal. Two easy methods are shown in Tables 2.3 and 2.3: tagging each word with a number (One-Hot Encoding, Table 2.2), and counting the frequency of terms in various text pieces (Bag-of-Words, Table 2.3). The data produced by either approach is high-dimensional and sparse (consisting mostly of zeros). Furthermore, there is a significant disadvantage to making use of such data. It does not imply that the two terms are interchangeable. A word like “cat” would sound just like “mat” if “jaguar” or “cheetah” were the word. Therefore, the model cannot transfer what it has learned about cats to the far less common term

“cheetah.” This is known as a lack of generalization power and it frequently results in subpar model performance.

**Table 2.2: One-Hot Encoding Example**

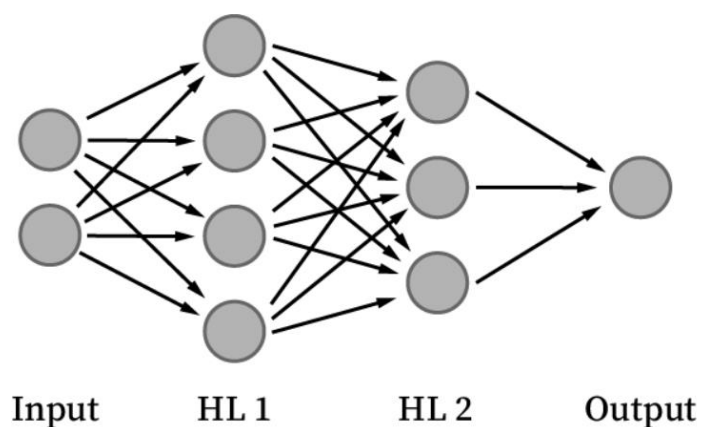
<b>Index word</b>	Deep	learning	is	fun	<b>Encoding</b>
1	1	0	0	0	(1000)
2	0	1	0	0	(0100)
3	0	0	1	0	(0010)
4	0	0	0	1	(0001)

**Table 2.3: Documents Representations using Bag-of-Words**

<b>Document</b>	Doc 1	Doc 2	Doc 3	Doc 4
<b>content</b>	<i>deep learning is fun</i>	<i>deep learning is machine learning</i>	<i>machine learning</i>	<i>Natural Language Processing</i>
deep	1	1	0	0
fun	1	0	0	0
is	1	1	0	0
language	0	0	0	1
learning	1	2	1	0
machine	0	1	1	0
Natural	0	0	0	1
processing	0	0	0	1

The lack of generalization can be solved by word embedding by representing words in a dense continuous n-dimensional vector representation. The vector distance between two words can show their semantic similarity. These word embeddings are often learned using neural networks. Once learned, they are transferrable and may be utilized in various contexts. The advantages of word embeddings over the previously described representations have led to their extensive use in contemporary NLP projects. The “distributional theory” underpins learned word embeddings (for more, see [31]). It argues that words with an equivalent frequency of occurrence in similar contexts are most likely

to have similar meanings. The two most well-known approaches for generating word embeddings are Word2vec by [32] and GloVe by [33]. Word2vec models, Continuous Bag-of-Words (CBOW) and Skip-gram, use a basic feed-forward neural network to predict a target word given its context in the case of CBOW or the context words have given a target word in the case of skip-gram. Although comparable to previous models, GloVe surpasses them by providing global word co-occurrence information in addition to local segment counts. Figure 2.1 displays a fundamental feed-forward network with fully linked layers for learning word embeddings within the context of word prediction. For this particular case, we first train the word embeddings in a projection layer and then use them in two hidden layers to replicate the probability distribution throughout the whole lexicon. Certain problems with input and output vectors of a fixed length may be solved well by a feedforward neural network. However, several NLP tasks lack predefined dimensions. Therefore, recurrent and convolutional networks may be used to circumvent the limitations of a typical network architecture.



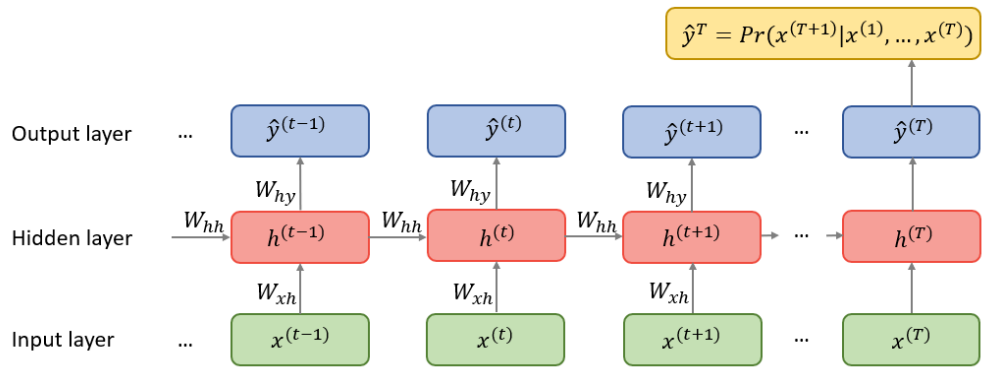
**Figure 2.1: An Example of a Fully Connected Feed-Forward Network**

### 2.3.3.2 Recurrent Neural Networks (RNNs)

The fundamental disadvantage of feed-forward neural networks is that they are predicated on a fixed length of input and output vectors. However, for many natural language issues, such as machine translation and speech recognition, defining appropriate fixed dimensions a priori is impossible. Other models that map one set of words to another set of words are required [34]. Recurrent neural networks (RNNs) are a type of neural network that was specifically designed to model sequential data such as text. RNNs process a sequence of words or characters  $x(1), \dots, x(t)$  by iterating through each element and recording information based on the preceding elements. This data is stored in the network memory as hidden states  $h(t)$ . The basic principle is straightforward: we begin with a zero vector as a hidden state (since there is no memory yet), process the current state at time  $t$  as well as the output from the previous hidden state, and feed the result into the next iteration [35]. A simple RNN is essentially a for-loop that reuses the values calculated during the previous iteration [36]. A traditional RNN structure can be seen in an unfurled computational graph (Figure 2.2). The gray square on the left represents a one-time step delay, and the arrows on the right show the flow of information through time [35].

Figure 2.2 illustrates that the model is shallow since each layer corresponds to a single parameter matrix. It is conceivable to extend this structure to a deep RNN, however, this is not an easy task given that each unit in an RNN is already represented as a nonlinear function of several units.





**Figure 2.2: RNN Network Layers**

### 2.3.3.3 Long Short-Term Memory Network (LSTM)

The recurrent neural network known as a Long Short-Term Memory (LSTM) network may learn to account for the importance of sequence order while solving sequence prediction tasks. This kind of behavior is necessary for solving difficult problems in areas like machine translation, and speech recognition, among others. As a subfield of deep learning, LSTMs are particularly challenging. Understanding what an LSTM is and how concepts like bidirectionality and sequence-to-sequence fit within the field may be challenging [39].

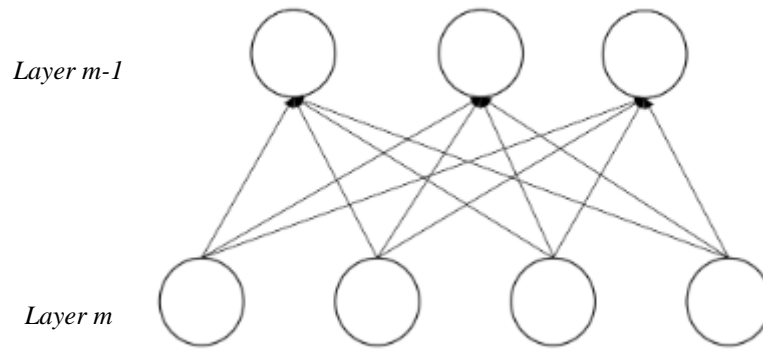
### 2.3.3.4 Convolutional Neural Networks (CNN)

No single algorithm used in machine learning or deep learning is limited to use in a single domain. Most successful algorithms can, with very minor adjustments, be used to provide excellent results in other domains. As is well-known, the field of computer vision makes extensive use of convolutional neural networks (CNN). Several academics have begun to investigate the use of convolutional neural networks for NLP after seeing their success in the image recognition domain. Even though early studies only focused on phrase classification tasks, CNN-based models have shown very substantial results,

demonstrating that the technique can be applied to several issues in natural language processing. Similarly, as was previously mentioned, the recurrent neural network (RNN), a type of sequence learning model, is one of the most popular deep learning models in NLP and sees extensive use in speech processing.

In recent years, one of the most prominent text processing techniques has been using word embeddings calculated using either the Word2vec algorithm or the GloVe algorithm as model input. Simultaneously, the vital efficacy of CNN in computer vision has been established. As a result, it seemed obvious that NLP tasks would require the use of CNN to word embedding matrices and the automatic extraction of features.

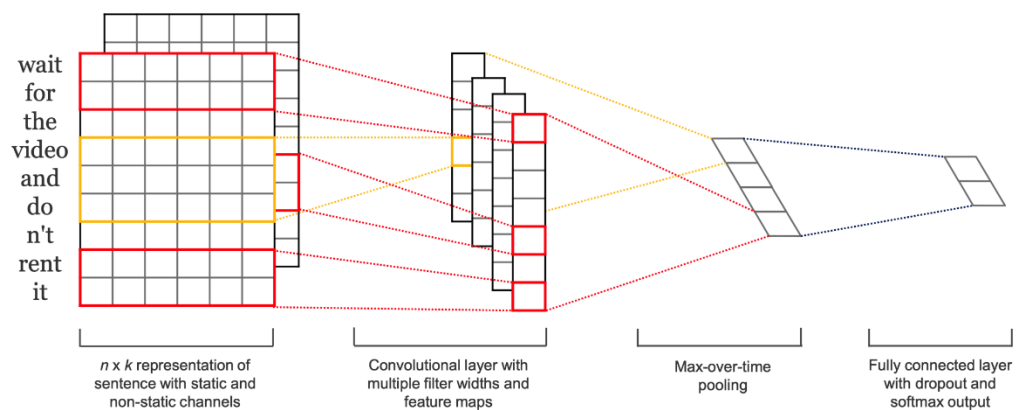
Figure 2.3 shows the unique structure of convolutional neural networks (CNNs) compared to classic neural networks. It consists of two layers, layer  $m-1$  and layer  $m$ , and illustrates the connections between neurons in these layers. In classic neural networks, all neurons in layer  $m-1$  are connected to all neurons in layer  $m$ , while in CNNs, only some neurons in layer  $m-1$  are connected to neurons in layer  $m$ . This creates a spatially-local correlation between nearby layers, which allows the network to better recognize patterns in images and other complex data. The figure also includes a diagram of a completely interconnected building, which is a representation of the traditional architecture of neural networks.



**Figure 2.3: Neurons Linking in the Convolutional Neural Network**

### 2.3.3.5 Sentence Classification by CNN

For tasks such as sentiment analysis, (Kim, 2014) developed an efficient and theoretically straightforward architectural paradigm [37]. As seen in Figure 2.4, a basic CNN design with a single convolutional layer is used, and the overall architecture consists of the following substructures:



Source: Y. Kim.(2014). Convolutional Neural Networks for Sentence Classification

**Figure 2.4: CNN Architectural Design**

1. **Sentence Representation:** The first step in representing a phrase is to assume that it contains  $n$  words, each of which has a corresponding representation in the  $k$ -dimensional word vector  $x_i$ ;  $\{i \in \mathbb{N} \mid 1 \leq i \leq n\}$  and  $x_i \in \mathbb{R}^k$ , where  $\mathbb{R}^k$  represents the set of  $k$ -dimensional real-valued vectors, where each element in the vector is

a real number. A sentence is therefore represented as  $X_{1:n} = X_1 \oplus X_2 \oplus \dots \oplus X_n$ , where  $\oplus$  is the concatenation operator.

2. **Convolutional Layer:** Let a filter denote as  $\omega \in \mathbb{R}^{hk}$ , which is used to a window of  $h$  words.

A feature map  $c = [c_1, c_2, c_3, \dots, c_{n-h+1}]$  can be generated by:  $c_i = f(\omega \times x_{i:i+h-1} + b)$  where  $b \in \mathbb{R}$  is a biased term. The feature map  $c$  is a sequence of activation values generated by applying a filter (or kernel)  $\omega$  of size  $h$  to a window of  $h$  words in the input sequence  $x$ . Specifically,  $c_i$  is the output of applying the filter  $\omega$  to the window of words  $x_{i:i+h-1}$ , adding a bias term  $b$ , and passing the result through a non-linear activation function  $f$ . The resulting feature map  $c$  has  $n - h + 1$  elements, where  $n$  is the length of the input sequence  $x$ . Each element  $c_i$  represents the activation of the filter at position  $i$  in the input sequence. Feature maps are a fundamental component of convolutional neural networks (CNNs) and are used to extract features from input data in various tasks such as image and text classification.

3. **Max Pooling:** Pooling operation has been applied for the respective filter to select the most important feature from each feature map  $\hat{c} = \max(c)$ , notice that one feature  $\hat{c}$  are generated by one filter, and these features will be passed to the last layer.

4. **Fully Connected Layer:** The selected features  $Z = [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_l]$  from the previous layer have been flattened into a single vector, to aggregate each of them and therefore a specific class can be assigned to it based on the entire input. In the given context,

$Z$  is a vector that represents the selected features from the previous layer after flattening. It is formed by concatenating the features  $\hat{c}$  from the previous layer, where  $\hat{c}$  is a modified version of the feature map  $c$ , and  $l$  is the number of filters used in the previous layer. The concatenation is done to combine the information learned by each filter, which is then used to classify the input into a specific class.

#### **2.3.3.6 Transfer Learning for NLP Tasks**

In traditional machine learning, a new model must be trained for each unique application. One way that we might tackle the challenge of learning a new activity is by transfer learning, which makes use of the previously labeled data from other similar tasks or domains. The model aims to accomplish a specific objective such as classifying an idiom sentiment, whereas the sourced knowledge is acquired during the next sentence prediction task. By avoiding the need to collect extensive amounts of training data in the target domain, reducing the amount of time spent on model training, and generally improving model performance, transfer learning offers various benefits over traditional machine learning. Given that there is a wealth of information about many texts beyond what is often found in training data, this is an extremely relevant issue in NLP challenges. Many language phenomena, such as long-term dependency and negation, may be captured and learned by a traditional NLP model using a large corpus. We may use this knowledge to retrain the model for a particular NLP task, like sentiment analysis. 38].

#### **2.3.3.7 Self-attention**

In the past, and to some degree now, recurrent neural networks using LSTM or gated recurrent units were the most popular models for language

modeling and machine translation [39]. One typical structure for these models is an encoding and decoding pair. The Transformer employs a creative architectural design for both the encoder and decoder but otherwise retains the standard encoder-decoder arrangement. The encoder has 6 Layers, each of which is divided into 2 sublayers. Self-attention in the first layer makes it possible for a transformer model to model the connections between all words in an input phrase simultaneously. This makes transformers more efficient than RNN and CNN-based models in modeling long-range dependencies in a phrase. Bidirectional Encoder Representations from Transformers (BERT) is a language representation model that was introduced by Devlin et al. in 2018 [9]. BERT is based on the Transformer architecture and it is pre-trained on a large corpus of text data using a masked language modeling task. Unlike traditional language models that use unidirectional context to predict the next word in a sentence, BERT employs a bidirectional approach, where it considers the context on both sides of a given word. This allows BERT to capture the meaning of a word in its entire context, resulting in better performance in various natural language processing tasks such as question answering, sentiment analysis, and text classification. BERT has become a significant breakthrough in the field of transfer learning for natural language processing and has been widely adopted in both academia and industry. To this day, BERT and its descendants remain the gold standards of transfer learning for natural language processing. Devlin *et al.*'s invention of Bidirectional Encoder Representations from Transformers (BERT) was motivated by the speed boost and the fact that “individual attention heads learn to perform different tasks” [9].

### 2.3.4 Classification using Pretrained Embedding

Word embedding takes a vast text corpus and extracts the semantic meaning of each word based on the context in which it is found as real-valued vectors in a space with fewer dimensions. In modern word embedding, a statistical method is used as the foundation rather than linguistics or modeling based on predetermined principles. For example, a real-valued vector with tens or even hundreds of dimensions accurately represents each word in the English language. The most common approaches to acquiring word embedding are as follows:

- **Learning from the ground up:** We first design the neural network's architecture, then train word embedding in conjunction with the primary task (e.g., sentiment classification). That is to say, we would begin with some random word embedding, and it would update itself along with the word embedding as they were added.
- **Learning by Transferring Knowledge:** Transfer learning is based on minimizing the time spent re-creating previously successful strategies. It allows transferring of information obtained or learned in some other work and utilizing it to improve the learning of another task connected to the first (the downstream task). In reality, one method for accomplishing this is, for the embedding component of the neural network architecture, we load some other embedding that was trained on a different machine learning task than the one we are attempting to solve and use those to bootstrap the process.
- One scenario where **transfer learning** excels is when there is little

training data available, and our data may need more sufficient to acquire task-specific embedding/features for our language. Using a word embedding that captures general aspects of the language might be advantageous from both a performance and time standpoint (i.e., we won't need to spend hours/days training a model from scratch to get comparable performance). What constitutes a good embedding depends much on the task at hand. For example, as the corpus's semantics vary between two tasks' objectives, the word embedding for the sentiment classification model may appear quite different from a topic-based document classification model.

In the scope of this research, we are mainly interested in the second learning method (Learning by Transferring Knowledge). Existing transformers give users access to thousands of pre-trained models that may be used for various applications. When we utilize a model that has already been trained to solve a specific task, we can retrain the model using unique or new data to solve a downstream task. This kind of retraining is known as fine-tuning process. Transformers may be fine-tuned using a variety of procedures and approaches as the following:

- We can use Layer-wise Learning Rate Decay (LLRD). The goal of LLRD is to apply different learning rates to each layer of the Transformer, or in the case of grouped LLRD, to the grouping of layers. To be more specific, higher layers need to have a more rapid rate of learning than lower ones.
- The learning rate plan may incorporate phases for pre-learning



or warming up. The learning rates increase linearly from 0 to the initial learning rates stated in the optimizer during the warm-up phase of a linear schedule with warm-up steps, and then they continue to decrease linearly until they reach 0.

- Re-initialize the top  $n$  layers of the Transformer, which encode information more specific to the pre-training task, is another option we can use. In most cases, we may use the transformer's pre-trained weights, since it has already been trained on a huge body of text data. However, occasionally we need to throw out some of these weights and re-initialize them during fine-tuning to obtain better performance. The selection of a suitable value for  $n$  is of the utmost importance, given that the quality of the results may start to deteriorate if more layers are re-initialized beyond the optimum threshold (the  $n$  value).
- The stochastic weight averaging (SWA) technique is another option for fine-tuning. This is a method of training for deep neural networks that uses a modified learning rate schedule. In addition, it maintains an ongoing average of the weights achieved after the most recent training session.
- The last approach to Transformer fine-tuning might benefit from performing more regular evaluations. The number of epochs controls how many times the learning algorithm will repeatedly analyze the whole training dataset. All of the samples in the training dataset will have been used to refine the internal model

parameters after one epoch. An epoch may consist of a single batch or many batches. Therefore, by doing frequent evaluations, we don't wait till the end of each epoch to check our model's accuracy; instead, we frequently check our model's accuracy after every x batch of training data within each epoch.

### 2.3.5 An Overview of Key NLP Models

**ELMO - Embeddings from Language Models:** Peters *et al.* (2018) introduced the deep, bidirectional LSTM model used in ELMO: to represent words [42]. This technique surpasses standard embedding approaches because it considers the words in their natural setting.

**ULMFiT - Universal Language Model Fine-tuning for Text Classification:** is a three-stage process that begins with pre-training the language model on a generic domain (like WikiText-103 dataset), then fine-tuning the language model for the downstream task, and finally fine-tuning the multilabel classifier such that it can classify each input sentence.

**BERT - Bidirectional Encoder Representations from Transformers:** Devlin *et al.* (2018) publish BERT, a paper written by members of the Google AI Language team. Proposing a bidirectional Language model based on a transformer was a major step forward in the field of natural language processing [9]. To overcome the limitations of one-way training, BERT proposes two novel pre-training objectives—a “masked language model” (MLM) and a “next sentence prediction” (NSP) task—based on the structure of the Transformer Encoder. For eleven natural language processing (NLP) tasks, BERT achieves state-of-the-art performance and its upgraded variations. Similarly, fruitful

efforts can be found in the work of “Albert” by Lan *et al.* (2019) [43] and “Roberta” by Liu *et al.* (2019) [44].

**GPT2 - Generative Pre-Training-2:** Researchers at OpenAI suggest a method called GPT2 [43]. In its most extensive configuration, GPT-2, a massive multilayer Transformer Decoder, employs 1.543 billion parameters. Although GPT-2 delivers state-of-the-art performance in a zero-shot scenario on 7 out of 8 evaluated datasets, it still underfits the newly created “WebText” dataset used to train it.

**XLNet:** Scientists from Google Brain and Carnegie Mellon University have suggested XLNet [44]. It takes inspiration from autoencoding (e.g., BERT) and autoregressive language modeling (e.g., Transformer-XL Dai *et al.*, (2019)) [47] but avoids their drawbacks. With the help of a permutation operation in the training phase, an autoregressive language model can learn to recognize both forward and backward contexts.

## **2.4 Data Augmentation Methods for Classification Problems**

Data augmentation refers to the processes that are used to increase the quantity of data. These processes may either include the creation of brand-new synthetic data based on the existing data or the addition of copies of the present data that have been extensively updated. Improving the variety of training data is one of the key goals of data augmentation techniques since doing so will aid the model's ability to generalize to new testing data. However, in contrast to the field of computer vision, where the augmentation of image data is a common practice, the field of NLP is still in its infancy. Because of the semantically invariant transformation (the message of an image is not altered by performing

simple operations on it, such as shifting its orientation by a few degrees or converting its colors to grayscale) augmentations quickly became an important toolset in the field of computer vision research [48].

To increase model generalization on downstream tasks, supplemented data is also anticipated to be different based on validity. The diversity of enhanced data is included here. According to the variety of their enhanced data, we may classify data augmentation techniques into three groups: paraphrasing, noising, and sampling.

- **Paraphrasing-based approaches:** Based on appropriate and constrained alterations to phrases, these approaches produce enhanced data with little meaning different from the original data. The enhanced data communicate information that is substantially close to that in the original form.
- **Noising-based approaches:** The validity is guaranteed by the noise-based approaches, which introduce discrete or continuous noise. The goal of these techniques is to increase the model's resilience.
- **Sampling-based approaches** can sample novel data within the data distributions that they have mastered. These techniques produce a wider variety of data and meet more downstream task requirements based on trained models and artificial heuristics.

### 2.4.1 Paraphrasing-based approaches

Paraphrases are methods to express the same information as the original form, which is a typical phenomenon in natural language [49-51]). Naturally, the creation of paraphrases is a good method for enhancing material. There are several degrees of paraphrasing, including lexical, phrase, and sentence paraphrasing (Table 2.4).

**Table 2.4: Summary of the Paraphrasing-based Data Augmentation**

<b>Augmentation Approach</b>	<b>Methods</b>	<b>Description</b>
Paraphrasing	Thesaurus	Substitute a word at random from the phrase with its synonym
	Semantic Embeddings	The initial word in the sentence of a phrase is replaced by the nearest neighbor in the embedding space
	Masked Language Model	Deep learning models that have been trained on a significant amount of text to do a particular job are known as transformers. To predict masked words based on their context, models like BERT were trained on the task of “Masked Language Modeling” [52].
	Machine Translation	This method relies on machine translation to retrain the meaning of a sentence by paraphrasing it. The common procedure follows the back-translation method by translating an English sentence to another language and then translating back the sentence to English.

## 2.4.2 Noising-based approaches

Unlike paraphrasing, noising-based approaches introduce light noise that suitably deviates from the original data while having no impact on the semantics [51].

### 2.4.2.1 Swapping

Swapping can be done at the word or sentence level. Generally, a little change in text order doesn't significantly affect how humans read it. However, the natural language's semantics are sensitive to it [53]. Therefore, an acceptable range of random word or phrase swapping can be employed as a data augmentation strategy.

### 2.4.2.2 Deletion

With this technique, phrases or words within a sentence are randomly deleted from a text. It was recommended by Wei *et al.* (2019) to delete individual words from the tweet at random with a probability  $p$  [54].

### 2.4.2.3 Insertion

This method involves the insertion of phrases or words into text in a haphazard fashion. Regarding the insertion at the word level, Wei *et al.* (2019) suggested picking a random word  $w_i$  in a tweet of  $n$  words  $T: \{w_1, w_2, \dots, w_n\}$ , that is not a stop-word, and then inserting a random synonym of that word  $s_i \in \{w_{i_{syn1}}, w_{i_{syn2}}, \dots, w_{i_{synj}}\}$  into a random position in the tweet and repeat this process to simulate the effect of inserting words at the word level [54].

#### **2.4.2.4 Substitution**

In this particular method, words and phrases are substituted with random string combinations. In contrast to the methods that have been outlined in the preceding paragraphs, it is common practice in this kind of paraphrasing to avoid utilizing words or phrases that are semantically close to the source text.

#### **2.4.3 Sampling-based approaches**

Methods that are based on sampling take into consideration the distribution of the data and sample newly gathered information from within it. They are constructed using trained models and artificial heuristics to satisfy the extra needs of downstream tasks, and they may be tailored to match the requirements of a given task. The techniques that are based on sampling are distinct from one another in that they are task-specific and need information about the task, such as the format of the data and the labels that are used to annotate the data. These kinds of tactics increase diversity while maintaining authenticity. Because of this, they are often more adaptable and difficult to work with than the previous two categories.

##### **2.4.3.1 SentMixup**

This method, rather than providing text in the natural language form, enhances samples by employing virtual embeddings [55]. Because it is based on the data that already exists, the sampled data that is included inside the virtual vector space may have labels that are distinct from those of the original data.

### **2.5 The Sentiment Analysis Task**

One definition of the NLP describes it as “an area of study and application that studies how computers might be used to comprehend and modify natural

language text or voice to accomplish valuable things.” The scope of the study of the NLP extends over a broad variety of academic fields, including “computer and information sciences, linguistics, mathematics, electrical and electronic engineering, artificial intelligence and robots, psychology, etc.,” [56]. This area of study has the potential to benefit several applications, including “machine translation, text summarization, information retrieval, question answering, speech recognition, and sentiment analysis” [56, 57].

While reasoning and analyzing unstructured data is a skill humans excel at, computers can distinctively process and analyze structured data [56]. The manual analysis of data obtained from social media is laborious and time-consuming, but it is necessary to ascertain the buried sentiments that are veiled in the transmitted words. However, because of the usage of non-standard, informal writing styles, shorthand, and metaphorical language, it may be difficult to computationally extract and categorize sentiments, which may not be implicitly represented in the message. Because of this, finding a mechanism to allow computers to rapidly process, evaluate, and interpret the emotion included in such a large amount of data is both a requirement and a demand. Fortunately, sentiment analysis has developed into a study field that is expanding quickly to meet these needs.

In recent years, social microblogging sites like Twitter have become reliable sources of news and data. Every year, the number of tweets predicted to be sent out rises by around 30% [58]. The fast growth of sentiment analysis is a direct result of the enormous size of the information landscape, and solving this problem has become an attractive study subject for academics all over the



globe. Specifically, the challenge is identifying and classifying hidden sentiments in a specific text among a large corpus of texts [17]. Sentiment analysis is a text-processing technique commonly known as “opinion mining” that aims to extract subjective information from a given data source using computational linguistics, classification algorithms, and NLP methods. The holistic goal of sentiment analysis is to assess users’ opinions or views on a specific topic, product, or service. Everyday use cases of sentiment analysis include product review evaluation, brand management, election campaign monitoring, and tracking customer comments on social media. The process of sentiment analysis involves mining a given text for thoughts or opinions and classifying their emotional tones into levels of sentiment polarity range. The most common task in sentiment analysis involves attempting to classify a text as either negative, neutral, or positive sentiment polarity.

Pre-processing the unstructured text by removing any noise or extraneous data is often the first step in the sentiment classification endeavor [56, 59]. According to Ahuja *et al.* (2019) “Pre-processing involves tasks such as tokenization, stop word removal, lower case conversion, stemming, removing numbers, etc.” [60]. The subsequent stage is to extract text features and represent them in various methods, such as count vectors, term frequency-inverse document frequency (TF-IDF), a bag of words (BOW) and pertained embedding. The last stage is to classify the data using lexical references or a machine learning/deep learning model [59].

### **2.5.1 Sentiment Analysis Levels**

The following are the primary levels at which sentiment analysis has been researched:

- A. **Document-level:** The sentiment analysis performed at the document level separates the overall document opinion into the different types of sentiment for a certain product or service. At this level, opinion documents are classified as either a positive, negative, or neutral polarity.
  
- B. **Sentence-level:** The sentiment analysis performed at the phrase level examines each sentence to evaluate if it represents a good, negative, or neutral view of a product or service. This category is suitable for ratings/reviews and comments/recommendations that the user has written themselves and which usually comprise just a single phrase or sentence.
  
- C. **Entity or Aspect-level:** The opinion mining and summarizing that takes place at the aspect level is based on the feature. Usually, we utilize this form of analysis when we need to know how reviewers think about a certain aspect or feature of the product. The classification process involves locating and selecting product attributes from the original data [61].

### 2.5.2 Sentiment Classification Methods

Counting the number of likes or votes on a post or tweet is the easiest way to figure out how people interact with something. To understand opinions, we need more than simple statistical measures to dig deeper into the message. Advanced linguistically aware processing methods, such as the conventional machine learning methods, deep neural network architecture, and pre-trained language models, have been used to reach this goal based on various text

processing methodologies, sentiment analysis may be loosely divided into two groups: lexicon-based and machine learning-based approaches. The former has the advantage of having a straightforward structure (corpus or dictionaries). However, building an accurate sentiment lexicon needs feature selection from massive and high-quality labeled data, which may be resource-intensive and time-consuming. Therefore, lexicon-based sentiment analysis has received minimal attention compared to machine learning approaches. However, successful machine learning needs correct data representation; hence, much effort has been put into the development of trustworthy feature extractors by using domain knowledge and rigorous engineering.

Deep learning models have attracted many researchers' attention to address the feature extraction problem. Deep learning algorithms can automatically extract meaningful text representations from data without needing feature engineering. Researchers suggest various deep learning-based algorithms for sentiment analysis, which outperformed machine learning-based algorithms in the sentiment classification task [62, 63]. Inspired by human brain structure and function, deep learning is a set of machine learning methods utilizing artificial neural network structure (multiple processing layers) to learn features or representations of an existing large amount of data. They offer a better capacity to represent knowledge and context.

Despite their tremendous success, these models struggle to extract complete features since they primarily depend on the specified training resources and the co-occurrence of the terms in the specific phrase. Deep learning models face difficulty extracting more comprehensive sentimental or

emotional features since much emotional information should be incorporated in the learning stage. Consequently, many researchers are attempting to integrate more emotional information and language knowledge into the deep learning models [64].

The primary techniques utilized in sentiment analysis are machine learning and deep learning methods [15]. Classification using deep learning is the best fit choice when a massive amount of data and less apparent features or patterns exists. Using a corpus to train a classifier for the sentiment classification task is essential from a machine-learning learning perspective (Lin *et al.*, 2020). Recently, deep learning has made great strides in text-processing tasks. One of the goals of deep learning is to unearth the building blocks of data representation at the sample level [65]. The insights and knowledge obtained via these learning programs are useful in this respect. The ultimate objective is to teach computers to read, comprehend, and classify information such as text.

When compared to traditional (un-)supervised machine learning algorithms, deep learning's complex neural network(s) perform far better in speech and image recognition tasks. When applied to the problem of sentiment analysis, neural network architecture is very effective. Many other types of neural network algorithms, such as bidirectional LSTM and CNN, are derived from the original neural network concept. A model based on the deep convolutional network approach was published by Conneau *et al.* (2016) [66] after Chen (2015) [67] successfully used CNN for the sentiment classification task and reported favorable results. Parameters for the models used in neural network approaches are often chosen at random at the outset before being

trained using optimization techniques like backpropagation and gradient descent [68]. However, before the development of pre-training approaches, the following problems were experienced when utilizing neural network-based deep learning for NLP tasks: At the start, deep learning models were not yet advanced enough to support complicated models. Second, there is a severe scarcity of labeled data for data-hungry deep learning models, and manual annotation is too costly to run on huge datasets. Therefore, researchers are devoting ever more time and energy to pre-training methods. For most NLP applications, deep learning techniques are now universally accepted as providing the highest level of accuracy [69-71]. However, Han *et al.* (2021) in [72] argue that deep learning models tend to over-fit when there are few labeled data sets and it takes a lot of time and effort to collect enough labeled data for deep learning models to be useful [70].

### **2.5.3 Sentiments of Idiomatic Expressions**

In the early days, techniques of keyword-based sentiment analysis were created to accomplish global sentiment classification using a bag-of-words frequency model. These techniques are still in use today. The problem with this classification strategy is that it does not take into account the emotional and structural connections that exist between words in the context of the specific setting[17]. Because of this, it often is unable to reveal the conveyed meaning in any way that goes beyond the literal or main sense. Instead, the focus of other research initiatives has been on finding ways to include local knowledge throughout the text. A good illustration of this is aspect-based sentiment analysis, which takes into consideration the polarity of aspects rather than only the existence of a collection of words inside a text. However, many of the

solutions that are now available do not take into account the contextual “intention” or meaning of words that are associated with the aspect. Additional factors, such as the non-standard use of a written language, idioms, slang, abbreviations, or the implicit discourse referring to an aspect indirectly affect the ability to accurately analyze the actual users' intention and the effectiveness of sentiment analysis systems. Linguistic features, such as idioms and other figurative language forms, may create semantic ambiguity and misinterpretation of the user's intention [17]. Rudra *et al.* (2017) provide evidence that idioms are used often on the Twitter platform [73]. They point to the fact that millions of people discuss idioms on social media platforms such as Twitter. They discovered that idioms were responsible for around 10% of Twitter trends over the course of ten months in 2014 [73].

In light of the difficulties outlined above, neither the lexicon nor the deep learning approaches provide a comprehensive answer to the problem of dealing with the sentiment classification of the always-evolving and changing slang, idioms, and abbreviations. Because certain idioms might have either a positive or negative connotation, we cannot always be able to ensure that the same attitude will be conveyed by an idiomatic expression. For instance, depending on the context in which it is used, the phrase “I had a blast” might mean either “had fun: = positive sentiment,” or “gone bonkers: = negative sentiment.” Because of this, the proposed sentiment classifier model has to use the assistance of keywords' surroundings (context) to acquire the various “meanings” or the true “sense” of an idiom. For the task of implicit sentiment classification, we will need to learn more text representation and need more semantic information to infer emotional inclinations from the text. Previous studies either only evaluated

discourse information without taking into account token dependencies, or focused on gathering features from a single sentence while disregarding contextual semantics [74]. Both of these approaches were flawed. Therefore, it is necessary to think about how to merge the two strategies into a single strategy to make up for the drawbacks of each approach.

## **2.6 Summary**

Opinion mining, also known as sentiment analysis, is the practice of analyzing online content for expressions of public opinion on a wide range of subjects, events, organizations, products, and traits [75]. We may discern these sentiments from the ocean of textual content available to us on the internet and social media. In addition, the advent of social media throughout the globe has spawned a new realm of knowledge that contains divergent perspectives on a broad range of modern issues [76]. Because of its widespread relevance in modern society, sentiment analysis has become standard practice in businesses, industries, and even governments and the political world. It's however expensive to analyze all of this textual "big data" that the web generates by hand, so solutions like sentiment analysis and opinion mining are in high demand smart. As machine learning and deep learning algorithms enhance natural language processing, sentiment analysis's usefulness in a wide variety of contexts continues to soar (NLP).

Customer behavior research, public response, user evaluations, and social media platform trends are just a few of the areas being examined by multi-language based sentiment analysis. As an example, the English language has amassed a wealth of information and a wide range of natural language

processing (NLP) tools [77], which has aided in the promotion and strengthening of the dependability of the decision-making process across many different areas [78].

In this chapter, we focused on how idioms are used to express and classify senses and feelings in sentiment classification challenges. We also looked into the textual elements and extraction and augmentation techniques used to determine and represent a text's emotional polarity. We focused especially on idioms, their characteristics, and their function in sentiment analysis. Finally, we surveyed the most cutting-edge methods for doing sentiment analysis using deep learning, as well as the most reliable ways for gauging their efficacy.



## CHAPTER 3

### RELATED WORK

In this chapter, we discuss the related work of the body of knowledge about sentiment analysis. In general, there are three principal methodologies for sentiment analysis. Namely, machine learning-based approaches, lexicon-based methods, and a hybrid methodology where a lexical and a machine learning method are combined to perform a seamless two-phased sentiment classification model [79-81]. The dictionary-based techniques and the corpus-based approaches are the two subcategories that are defined by researchers as being part of the lexicon-based methodologies. The dictionary technique classifies sentiments by utilizing a predetermined lexicon of terms, such as those that may be found in WordNet or SentiWordNet; Lexicons are compiled from the entirety of the document, and then any online thesaurus may be utilized to find synonyms and antonyms to extend further that lexicon [80]. Comparatively, the corpus-based analysis approaches focus on statistical data analysis, like the k-nearest neighbors clustering algorithm. Therefore, it does not need a preset lexicon.

Deep learning or conventional machine learning algorithms can be used in machine learning-based sentiment analysis. Conventional supervised classification has gained popularity due to its outstanding results; nonetheless, the primary disadvantage of supervised methods is that they are domain dependent. To avoid these issues, unsupervised methods are introduced to

overcome domain dependency. In addition, supervised methods require manual feature engineering to train the model. In contrast, deep learning has become dominant and surpassing conventional machine learning, because it can learn from text without needing the manual feature engineering process [80].

The majority of the strategies for unsupervised sentiment classification may be categorized using generative models [82-85] and lexicon-based algorithms [86-89]. The core of this research is to utilize an English Idiomatic Lexicon in combination with a deep learning approach to cover the unmet part in sentiment classification of Twitter data having idiomatic expressions.

### **3.1 Sentiment Classification using Machine Learning**

Despite the widespread use of lexicon-based sentiment analysis, the primary focus of the vast majority of sentiment classification research is on machine-learning methods [90]. Traditionally, the sentiment classification task has been supervised, requiring big datasets manually annotated by domain experts and used to train the learning algorithms. The phase of feature engineering in supervised-learning approaches may be the most important. Creating and developing features using data from the knowledge domain is essential for efficient classification. In text classification, feature creation, transformation, extraction, and selection are used to train and test/evaluate the classifiers on that specific domain. In testing or evaluation, the newly or previously unknown data is subsequently classified using the prediction model produced by the learning model.

Support Vector Machines and Naive Bayes supervised classifiers have been frequently utilized in the past to address sentiment analysis. In their paper,

Dhande & Patnaik recommend using a combination of Neural Network and Naive Bayes classifiers for sentiment classification tasks [91]. In this approach, a neural network is used to explicitly express the relationships among the features of words. This is done to circumvent the notion of attribute independence that is the foundation of the Naive Bayes classifier. It has been demonstrated that the resultant Naïve-Bayes-Neural classifiers attain promising accuracy when using a straightforward unigram representation of text messages.

Yue *et al.* (2019) present various machine-learning tools used to classify the sentiment of data derived from social media platforms [92]. The authors weigh the advantages and disadvantages of each approach. In supervised learning, the rule-based approaches are described as those “generate descriptive models,” and are simpler to understand. However, they claim that they are only useful when datasets are imbalanced [92]. In addition, they argue that the decision tree method is the easiest to implement but it suffers from the overfitting problem. In their experiment, SVM achieves the highest accuracy, yet it’s slow to train and computationally expensive. They also claim that the regression-based models suffer from the under-fitting problem even though they have a reasonable computational cost and are considered to be easy to understand and implement. However, they kept an open question about the complexity and the performance of the statistical models depending on the method they implement. On the ensemble methods, they record that this technique can overcome overfitting and perform better in generalization to produce high-performance predictive models. However, ensemble modeling is seriously hard to analyze and computationally expensive. They conclude that neural networks are slow but able to handle noisy data even though they produce

low accuracy and are considered to be computationally expensive. They point out that the benefits of deep learning models include the ability to handle deep architecture, analyze big datasets, and allow multi-task learning. However, deep learning models are data-hungry, intricate to interpret, and computationally expensive.

To improve the accuracy of an SVM in classifying sentiment polarities in tweets, Chikersal *et al.* (2015) suggest including a lexicon of emoticons and opinion words in the training dataset [93]. Khan *et al.* (2016) presented a similar approach by utilizing the SentiWordNet lexicon in conjunction with a support vector machine (SVM). The SVM was trained using a feature-weighting method that relied on pointwise mutual information [94].

Samal *et al.* (2017) studied and evaluated seven supervised machine-learning methods for sentiment analysis of movie reviews. According to their findings, linear SVM is the only classifier capable of classifying a large number of movie reviews with perfect precision [95]. The primary limitation of this study is the tiny size of the used datasets, which is ideally suited for supervised machine-learning approaches. Therefore, uncertainty surrounds the system's performance on huge datasets.

Suhaimi and Abas examined 305 research papers that utilized supervised machine learning to solve the sentiment analysis task. They screened the papers based on the selection criteria and data extraction processes they described, and 61 research were ultimately chosen [96]. They found that supervised learning has mostly been employed in classification studies for the healthcare and medical industries, as well as for spam text classification. According to them,

the most effective classification algorithms are SVMs and Artificial Neural Networks (ANN).

A thorough and detailed survey was provided in [97]. The authors contrast various machine-learning approaches and a lexicon-based sentiment analysis approach. The least accuracy method achieved 38.45% points. However, the accuracy of the linear Support Vector Classifier reaches 100% when working with big datasets. They conclude that the hybrid strategy proved to be the most effective answer because it includes the benefits of both techniques.

Even while several sentiment analysis problems have been handled using traditional supervised learning approaches, the bulk of contemporary work is trending toward an alternate interpretation of the topic. Current research focuses on creating diverse representation spaces by using word embeddings [98] to train conventional learning models (99-102). Severyn and Moschitti presented a new method for initializing the weights of a CNN [99] using word embeddings to ultimately train an appropriate softMax sentiment classifier [100]. They were successful in combining supervised learning on the data that was supplied with word embeddings to produce a rich language model. Supervised learning and rich language models are both necessary components for sentiment analysis. The authors of [101] also depicted an architecture of a supervised machine and a rich language model.

Even if the approaches described above represent a big step toward the creation of resilient systems, there is no widespread agreement over which tactic needs to be used to solve a specific issue in a certain field [102, 103]. Within the realm of studies about the classification of emotional reactions, there is not

one classification method that consistently surpasses the others. Recent research [104-109] has focused on investigating the ensemble learning paradigm as a potential approach to solving this issue. It is the goal of ensemble methods to boost the efficiency of baseline classifiers by integrating the results of several individual classifiers into a single model.

### **3.2 Feature Selection and Representation**

In machine learning, feature selection is crucial since it not only shrinks the feature space but also offers a less redundant feature subset which can boost classification accuracy. Thus, for sentiment classification, both conventional and sentiment-oriented feature selection strategies have been investigated. A comparison of the four procedures (Mutual Information, Information Gain, Chi-Square Test, and Document Frequency) reveals that Information Gain performs better than the other three traditional feature selection techniques when it comes to classifying the sentiment of Chinese texts [110]. In the dataset of movie reviews, a similar conclusion was noted by Sharma and Dey (2012) in [111]. Wang *et al.* (2011) also show that Information Gain may be improved by feature selection using Fisher's discriminant ratio [112]. Intriguingly, focusing on feature engineering has been conducted to improve sentiment analysis by addressing the problem of sentiment classes having ordinal correlations as opposed to having no apparent links [113]. Similar to this, a genetic algorithm was developed for multilingual sentiment classification [114], and a simple embedding model to generate latent features was applied to find terms with significant intra-similarity and inter-differentiation [115].

Other researchers focus their attention on various approaches to feature weighting. Word frequency schemes, which are often employed in information

retrieval (IR), have been the subject of several research. The findings indicate that these schemes may be altered to enhance the performance of the supervised sentiment classifier [114]. It should be emphasized, nonetheless, that the TF-IDF assigns term weights by compiling information from the whole corpus rather than giving much consideration to class labels. Therefore, the Delta-TFIDF was developed, in which the total term weight is determined by the difference between the weights of two documents belonging to the same sentiment class [117].

According to a ground-breaking study by Pang *et al.*, sentiment classifiers in supervised learning provide poorer accuracies in contrast to standard text classifiers when using the one-hot encoding (a binary vector representation of words) [118]. This shows that topic categorization is easier than sentiment classification in terms of difficulty. This gap arises, in part, because sentiment is communicated in more nuanced ways than the basic representation can fully reflect. As a result, more complex linguistic features were researched to improve the representation, such as using part-of-speech tags to distinguish between different uses of the same word, making use of positional information about terms in a phrase or a sentence, or using n-grams rather than one-hot encoding, which only checks whether a word is present or not [118]. The experiments proved that these representations are inferior to the fundamental one-hot encoding form in terms of effectiveness. These discoveries have increased the demand for feature engineering methods that are more advanced. In addition to the above features, syntactic relations, feature subsumption hierarchies, appraisal groups (such as “very good”, and “not funny”), and syntactic relations have all made major contributions to feature discovery [119-121].

The bag-of-words (BOW) model, which is frequently used in supervised machine learning for text representation, is deficient in several term interdependencies that are crucial for classifying sentiments (such as negation and intensification) [122]. To overcome this issue, relevant qualities are introduced to add contextual information for a given term (for example, “neg bad” replaced a negated “bad” and “int bad” represents the intensified “bad”) [123]. Additionally, the BOW model is unable to accommodate word variants like polysemy, antonymy, and synonymy, which are widely employed in semantic indexing strategies [124]. These methods project documents or expressions from other low-level representations (such as n-grams) to high-level semantic concepts. The spaces of particular words and latent (hidden) semantic ideas have both been presented to change text representations. For instance, the Latent Semantic Indexing (LSI) method identifies latent semantic relationships between words that are not apparent in a low-level representation of documents. LSI employs singular-value decomposition (SVD) to project low-level space to high-level semantic concept space by calculating word co-occurrence patterns that are present in the corpus [125]. In another, work, a more straightforward method is to infer semantic similarity from statistical assessments of term co-occurrence inside texts [126]. However, these approaches do not consider the documents' membership in any particular classes. The representations of emotional states that come up as a consequence of their application are not the most useful ones for making classifications. To circumvent this issue, Sani *et al.* (2014) came up with the idea of using supervised sub-spacing (S3) for supervised semantic indexing of texts [125]. The technique generates an individual sub-space for each class, which enables



it to give a supervised way of extracting term-relatedness [125]. According to the findings of the research, the performance of S3 is superior to that of other existing classifiers that make use of the BOW scheme.

### **3.3 Word Embedding**

Word embedding is a kind of embedding space in the sense that it is not a vector space but an  $n$ -dimensional vector space. It can be compared to the geometric concept of a base dimension, where the width is taken to be “1”. The motivation behind word spaces arises from the study of language data distribution, which is naturally modeled as vector spaces. The key idea behind word embeddings is to use this word distribution information to encode words into low-dimensional vectors, and then distribute these vectors back into a word representation space (e.g.,  $n$ -dimensional). This way, the relationships between words are described as a combination of their embedding vectors (i.e., a dot product). In other words, the idea is to represent each word as one point in an  $n$ -dimensional vector space of words, where each dimension corresponds to the frequency that the word appears in a sentence. Thus, when people read a sentence and decode it, the information contained in each word is represented by its vector.

Word embeddings, such as Word2Vec, are used as features in several NLP tasks. To learn the embeddings of individual words, we may utilize either neural networks or matrix factorization. Word2Vec is a neural network prediction model that uses textual information to efficiently learn word features (embedding vectors). It combines the Skip-Gram (SG) model with the Continuous Bag-of-Word (CBOW) model. The SG model deduces the meaning of the target word (in this case, “crypto”) from the words that immediately

precede it in the sentence (for example, “The \_ currency costs a lot”). The CBOW model deduces the meaning of the surrounding words based on the target word alone. The CBOW model performs statistical analysis on the full context as if it were a single observation, which results in a considerable amount of distributional information being smoothed away. When used to relatively small datasets, CBOW performs quite well. Nevertheless, the Skip-Gram model is preferable for usage with bigger datasets because each pair (context, target) is handled as if it were a new observation. Another popular method of learning is GloVe. It is an unsupervised learning technique that trains solely on the nonzero elements of a word-word co-occurrence matrix to collect statistical information as a vector representation for words.

The BERT model is an illustration of contextual embedding that is skillful and accurate in representing words inside phrases [42, 127]. BERT is specifically a version of the Transformer architecture [41], which has raised the bar on several tasks, including language modeling and machine translation [128, 129].

### **3.4 Lexicon-based sentiment analysis**

A sentiment lexicon is a comprehensive collection of words and phrases in a language along with their associated emotional connotations, such as positive, negative, or neutral. By using a sentiment lexicon, sentiment analysis tools can automatically classify text according to the emotional tone conveyed by the words used. This saves a lot of time and effort, and can provide valuable insights into how people feel about a particular topic or product. Therefore, having a sentiment lexicon that catalogs the emotions and sentiments associated with each word in a language is an invaluable resource for sentiment analysis

[130]. Several researchers use sentiment lexicons as a training feature in supervised machine learning algorithms, or as an input for unsupervised sentiment models [131]. Therefore, machine-learning techniques and rule-based approaches depend heavily on lexicons like these [80]. In some cases, the Lexicon-based approach is superior to the traditional supervised machine-learning approach in terms of both accuracy and efficiency [132]. The authors refer to this success due to the development of a comprehensive lexicon and an effective Urdu Sentiment Analyzer.

As discussed in [130, 132], a sentiment lexicon is a set of words that are organized according to whether they convey a positive, neutral, or negative emotional tone. Sometimes called “polar words” or “opinion words,” these terms express strong, often opposing, views. Positive emotions may be communicated via the use of a variety of adjectives such as wonderful, stunning, and majestic. On the other side, words like “awful,” “lousy,” and “bad” are examples of words that communicate a negative attitude.

The best way to handle complex data for analysis is to use sophisticated lexicons that account for words’ subjectivity or objectivity, context, and intensity [134]. As an example of terminology, if a term is considered a positive sentiment, then consider how positive it is; there is a distinction between excellent, outstanding, and extraordinary. Unfortunately, there are not many easily accessible internet emotion or sentiment lexicons for most languages [135, 136]. The words and sentiments they are associated with are listed in a single file in certain sentiment lexicons (both negative and positive). A polarity indicator, such as (positive, negative), (0, 1), or (1, -1), is added to the second

column of this list. The words or concepts themselves are located in the first column. Some sentiment lexicons include a structure that includes the idea of “sentiment intensity,” while others offer the part of speech (POS) for each word [137]. Both types of lexicons may be used to analyze sentiment. Table 3.1 displays some values for the strength, length, POS, stemmed, and polarity of a sample word list from the MPQA lexicon [138].

**Table 3.1: Sample word list from the MPQA lexicon**

<b>word1</b>	<b>type</b>	<b>len</b>	<b>pos1</b>	<b>stemmed1</b>	<b>priorpolarity</b>
agonize	strongsubj	1	verb	Y	negative
agonizing	strongsubj	1	adj	N	negative
agonizing	strongsubj	1	anypos	Y	negative
agonizingly	strongsubj	1	anypos	N	negative
agony	strongsubj	1	noun	N	negative
agree	strongsubj	1	verb	Y	positive
agreeability	strongsubj	1	anypos	Y	positive
agreeable	weaksbj	1	adj	N	positive
agreeable	strongsubj	1	anypos	Y	positive
agreeableness	strongsubj	1	anypos	Y	positive
agreeably	strongsubj	1	anypos	Y	positive
agreement	weaksbj	1	adj	N	positive
agreement	weaksbj	1	noun	N	positive

Some researchers, such as Liu [130], have split sentiment lexicons into two separate files, with the first file including terms that convey a positive sentiment and the second file containing terms that convey a negative connotation. The emotional orientation, represented by the polarity value, may be communicated in many different ways.

Valence Aware Dictionary and Sentiment Reasoner (VADER) is one of many well-known lexicons [139]. It was explicitly developed to evaluate the sentiments expressed in a text [139]. It is frequently employed to assess the sentiment of social media posts. For VADER, polarity and intensity are both equally significant. Table 3.2 shows a sample of the Vader Lexicon.

Another lexicon is the SentiWordNet [140]. SentiWordNet is an enhanced extension of WordNet [141]; the synsets are logical collections of cognate synonyms made up of verbs, nouns, adjectives, and adverbs. WordNet is a lexical database of words identified in the NLTK corpus and is based on the relationships between terms. This aids in determining the polarity information relevant to the specific word occurrence.

**Table 3.2: Sample Keyword Annotations in VADER Lexicon**

<b>Token</b>	<b>Mean</b>	<b>SD</b>	<b>Raw-Human-Sentiment-Ratings</b>
overstatement	-1.1	0.7	[-2, 0, -1, -2, -1, 0, -1, -1, -2, -1]
party	1.7	0.78102	[3, 2, 2, 1, 3, 2, 1, 1, 1, 1]
respective	1.8	1.16619	[2, 2, 3, 0, 1, 3, 3, 1, 0, 3]
scared	-1.9	0.7	[-1, -1, -2, -3, -2, -3, -1, -2, -2, -2]
sucked	-2.0	0.89443	[-2, -2, -1, -1, -1, -3, -4, -2, -2, -2]
troublesome	-2.3	0.78102	[-3, -2, -3, -2, -3, -3, -1, -2, -1, -3]
weak	-1.9	0.7	[-1, -3, -2, -2, -3, -2, -2, -1, -2, -1]

The majority of academics rely on sentiment lexicons that can be accessed in English and have been manually built for increased precision. The time and energy required to construct new non-English sentiment lexicons have been greatly reduced with the help of the existing English sentiment lexicons [142]. The SenticNet, SentiWordNet, and Opinion Lexicon are a few examples of well-known English sentiment lexicons. The purpose of creating these lexicons was to improve the accuracy of sentiment classification [202]. One such lexical resource utilized for this purpose is SentiWordNet, which is freely accessible to the public. It is constructed by assigning each synset in WordNet to a sentiment

class of positive, negative, and neutral labels. SentiWordNet assigns a numerical value between 0 and 1 to each phrase to show how important the words in the phrase are [140]. SentiWordNet, similar to other lexicons, does, however, have some noise in it since not all of the polarity values that are assigned to the words are true. This is the reason why SentiWordNet does not have a perfect accuracy rate. In addition, certain words do not possess a polarity value, but other phrases possess values that are in direct opposition to one another [136]. For example, in SentiWordNet, some entries may be found under two separate polarity headings and have been given a positive label in the first entry and a negative label in the second entry. By classifying the polarities of words according to the POS to which they belong (nouns, adjectives, verbs, and adverbs), SentiWordNet is also able to assign polarity at the syntactic level. The letters “n,” “a,” “v,” and “r” are used, in that order, to denote this [143]. Syntactic polarity assignment refers to the stage at which a sentence is constructed. Just like SenticNet, which is a public resource, SentiWiki is a reference tool for the public. SenticNet was created by making use of artificial intelligence and semantic Web technologies, and it is dependent on an innovative dimensionality-reduction method to establish the polarity of notions that are thought of as belonging to common sense [144].

The benefits of lexicon-based or rule-based approaches are that they are primarily utilized as tools for the unsupervised approach to eliminate the need for prior training. Additionally, they generally demonstrate speedier execution. The rule-based methods may successfully handle fewer problem cases compared to machine learning-based approaches. However, the latter approaches require an extensive dataset to train on. Additionally, if the correct

vocabulary is used on the exemplary domain instance, the accuracy of the rule-based technique is typically quite excellent, with consistent results. Finally, rule-based approaches exhibit lower risk, have undergone testing, and are widely used. The primary drawback of rule-based approaches is that they are domain-specific; for example, VADER performs better in cases of the social media domain [139].

The rule-based approaches have the additional drawback that they cannot learn and the rules must be initially created with the assistance of an expert. Additionally, adding or amending the rules necessitates subject-matter specialists and requires arduous human labor before being included in the completed product.

### **3.5 Approaches to building sentiment lexicons**

#### **3.5.1 Dictionary-based lexicons**

Using this method, a dictionary is compiled by selecting a few words at random as seed entries. After that, we utilize an online dictionary, thesaurus, or WordNet to extend the dictionary by adding synonyms and antonyms of the terms in question. In lexicon-based classification, documents are given labels by comparing the number of terms that come from two opposing lexicons, such as positive and negative sentiment lexicons.

##### **A. Semantic relationships approach**

The relationship approach makes advantage of the preexisting semantic relationships between words in a lexicon or dictionary to build atop a small core vocabulary (seeds) [145].

##### **B. Predefined lexicons approach**

The overarching goal of this approach is to enhance the level of precision achieved by sentiment analysis by integrating many predefined lexicons into larger, more comprehensive lexicons. This is particularly helpful for languages like Arabic, which have a dearth of vocabulary resources. The merging may take many different forms, including the combination of numerous lexicons written in the same language or the translation of several lexicons into another language before the merge is performed.

### **C. Limitations**

- These methods provide general-domain lexicons of emotions, which may seem less accurate in domain-specific contexts.
- There aren't a lot of social networks' specific (SNS) terms or slang in sentiment lexicons. Due to the lack of support for dialects and informal/slang terms, dictionaries are unable to handle them [146].

#### **3.5.2 Corpus-based lexicons**

A corpus is a large database of material that can be read by a machine, such as online discussions, academic papers, reviews, and more [27]. There are statistical methodologies and techniques for establishing semantic connections that may be used to build sentiment dictionaries. Statistical algorithms may use large corpora to build a new polarity-based sentiment lexicon by assessing the frequency of items inside a specific class. The second method constructs a sentiment lexicon through the semantic connections between words in a massive



corpus [27].

### 3.5.2.1 Frequency-based method

Formulas and methods from the field of statistics are utilized to calculate the number of times specific words appear within a provided polarity. This technique is based on the premise that positive terms are more likely to be found near other positive terms and that the opposite is also true. Point-wise mutual information (PMI) is a widely used statistical metric for distinguishing the link between terms of a corpus that are to be classified into distinct polarities.

Following is the definition of the PMI between two words,  $w_1$  and  $w_2$ .

$$\text{PMI}(w_1, w_2) = \log_2 \left\{ \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \right\} = \log_2 \left\{ \frac{C(w_1, w_2) * N}{C(w_1)C(w_2)} \right\}, \quad (3.1)$$

where  $C(w_1, w_2)$  represents the frequency of  $w_1$  and  $w_2$  when they appear together, whereas  $C(w_1)$  and  $C(w_2)$  computes the frequency of  $w_1$  or  $w_2$  as they appeared independently from each other.

### 3.5.2.2 Graph-based method

This technique utilizes the interconnectedness of words in a huge corpus to generate potential synonyms for existing words (seeds). When there is no difference in the amount of edging between two words, it suggests that they have a similar meaning or polarity of sentiment.

### 3.5.2.3 Limitations

- Corpus-based lexicon construction is complicated and time-consuming since many languages lack adequate data pre-processing techniques and or

online corpora

- To acquire sufficient accuracy when building sentiment lexicons through corpus analysis, a large corpus is needed; the resulting lexicon has few terms and often only applies to a specific domain. Therefore, it cannot be relied upon for analysis in any other field.
- Lastly, an annotated corpus is required by certain techniques. Therefore, more data annotation is needed before analysis can commence.

### **3.5.3 Human-based computing lexicons**

#### **3.5.3.1 Crowdsourcing**

Lexicon annotation by crowdsourcing service is the practice of obtaining sentiments and polarities of word or phrase lists by enlisting the services of a large number of people, either for monetary compensation or for free, most frequently through the use of the internet. Mohammad *et al.* (2013) utilized the crowdsourcing service to build a lexicon by annotating words by their emotion and sentiment polarity [147]. Williams *et al.* (2015) created a comprehensive sentiment lexicon of 580 idiomatic expressions using a web-based crowdsourcing service [148]. Other researchers proposed a game-based approach to creating a sentiment lexicon using online users.

Human-created sentiment lexicons tend to be more precise than automated ones [149]. However, creating such lexicons is labor-intensive, time-consuming, and costly. To mitigate these obstacles, several researchers

advocated for a gamified method whereby humans would be responsible for labeling texts with their emotions. The Tower of Babel [149], Like it! [150], and Tsentiment [16] are only a few examples.

### **3.5.3.2 Manual-based approach**

Linguists or other experts with deep knowledge in a certain field use this approach to create lexicons. Using this strategy, Abdul-Mageed *et al.* (2014) create a sentiment lexicon and use it for sentiment classification of Arabic social media data [151]. Trakultaweekoon and Klaithin create a web-based sentiment tagging tool called SenseTag to facilitate the annotation process of sentiment lexicons [152]. The tool was trained with the assistance of manual annotations that were supplied by linguists. These linguists labeled each word in texts that were chosen at random.

### **3.5.3.3 Limitations**

The human-based lexicon creation has a reputation for being time- and resource-intensive. Even though, a significant number of researchers continue to make use of this approach, especially for languages that have a restricted number of lexical resources.

### **3.5.4 Idiomatic lexicon-based sentiment analysis**

Rudra *et al.* (2017) attempt to demonstrate the significance of idioms on Twitter. They illustrate that millions of individuals use idioms, even though the same idiom may be used regardless of the topic or area in which they participate in a discussion [73]. They found that idioms account for around 10% of Twitter trends recorded during the first ten months of 2014 [73].

Early studies of sentiment analysis relied on an idiomatic lexicon for data classification. One of the first instances of this technique may be found in the work of [153]. To develop a method for classifying customers' sentiments about a product, the writers relied on a sentiment lexicon. To recognize and extract sentiment patterns from the text, they employed a sentiment word and idiom lexicon among other features. There are over a thousand idioms in English that they painstakingly collected. The authors noted that while it is laborious to compile and annotate idioms, the bulk of them express powerful emotions.

Shudo & Tanabe (2010) have published a comprehensive dictionary of Japanese multiword idioms. The clichés and common idioms in this lexicon are joined by phrases that are almost but not quite idioms. The dictionary's extensive coverage of alternate notations and derived forms makes it applicable in a broad variety of contexts [154]. However, this is a lexicon, and as such, illustrates the conceptual foundation of semantics and this prevents doing direct sentiment analysis on the data.

Mudinas *et al.* (2012) provide a set of tools for sentiment analysis (PSenti) [155]. To identify extreme opinions and quantify their intensity in online evaluations, they proposed a hybrid method that blends lexicon-based techniques with machine-learning approaches. Emotional words, 116 emoticons, and 40 English idioms are used in the hybrid method to attain high precision. Polarity is estimated by giving each emotion pattern a score between [-1, 1], with emojis receiving a score of [-2, 2] and idioms receiving a score between [-3, 3].

An unsupervised sentiment classifier was used to search for and extract

Chinese idioms from the text [156]. To compile the idiomatic sentiment lexicon, they combed through more than 24,000 idioms, selected around 8,000 samples from those idioms, and assigned each of those idioms a positive or negative orientation. By employing three publicly accessible Chinese product evaluations (for a book, a hotel, and a notebook PC), the authors analyze the efficacy and lexicon size of their classifier and suggest that using idioms improves classification accuracy.

AIPSeLEX was proposed by Ibrahim *et al.* (2015) as an idiomatic sentiment lexicon of contemporary Egyptian and Arabic dialects [157]. They describe the time-consuming work that went into creating the AIPSeLEX lexicon, which has 3,632 idioms and proverbs. There would be an increase in precision in the analysis of emotional states, according to their study, if idioms were included as a differentiating trait. Using merely a cosine similarity and a Levenshtein distance, they were able to properly classify the data into distinct sentiments.

By employing idioms as characteristics for a sentiment classification query, Williams *et al.* (2015) have shown that the overall performance of sentiment analysis is much improved [148]. They produced a lexico-semantic resource with 580 idioms annotated with the positive and negative sentiments they convey. They also used a database of regional grammar to spot all instances of these idioms in the text. They used crowdsourcing to annotate the lexico-semantic with the proper polarity. Idiom polarity acquisition is not automatic, but this method works well. Idioms are a rarity, which makes studying their function in the analysis of sentiments all the more challenging. Since the corpora

that are commonly used for testing algorithms for sentiment analysis have imbalanced usage of idioms, it is impossible to generalize the results of the idiom research. The major downside of this approach is the length of time needed to manually develop lexico-semantic criteria for detecting idioms and the polarity of those idioms.

In [158], Spasić *et al.* present an intriguing subject of investigation. In addition to presenting criteria for recognizing idioms in text, the authors also provided a way for automatically developing lexical semantics for sentiment polarity classification tasks. Early findings showed that this basic technique (combining idiom and phrase polarity) greatly improved sentiment analysis results; nevertheless, this approach often favors adopting the idiom's polarity above the polarity of the sentence and does not guarantee optimal performance [158,159]. Instead of this naive technique, we offer an automated feature integration mechanism that retains the “positional context” of an idiom within the original tweet or phrase. To enhance the accuracy of sentiment analysis via word disambiguation, Chen *et al.* (2021) proposed a method for labeling neural networks to better recognize Chinese metaphors [160]. They conclude that figurative language is more effective in evoking an emotional response than factual language. Synthesizing and engaging with the source and target semantics in metaphors may result in emotive content. Although the publication doesn't specifically address the issue, this methodology may be useful for tasks that need sentiment classification. The authors of [161] show how to generate a sentiment corpus and how to expand upon it by using a lexicon of idiomatic phrases of emotion. They determined the emotional tone of idiomatic phrases, tested them on a corpus of over a hundred sentences, and established a cutoff

point for including a phrase in the appropriate emotional bucket. They found that almost half of the idioms offered reliable sentiment assessments. The fundamental problem with this study is that the idioms' polarity strength is not taken into account when estimating the idioms' sentiment, instead relying only on the surrounding text. They find that it may be more difficult to attribute a sentiment to an idiom on its own and propose the need for a lexicon that takes the phrase's context into account. In this thesis, however, we propose an expansion approach to compute the overall emotion of a tweet while also taking into account the polarity of the phrase itself. It has been found in other studies like [162] that an enhanced version of BERT-like transformers can be utilized to create a hybrid model for idiom and literal meaning recognition. While idiom discovery is their major focus, they also demonstrate how BERT-like transformers may be fine-tuned to extract idiomatic meaning. Using an expanded version of PerSent, a manually labeled sentiment lexicon, Dashtipour *et al.* (2022) suggested a method to extract and classify the sentiment of Persian text containing idioms [163]. To determine the overarching feeling of a piece of Persian literature including an idiomatic statement, they used a variety of classification methods. Our approach is novel in that we claim the lexicon may be automatically annotated and then used in the deep learning classifier.

In [164], the authors Hwang & Hidey (2019) consider a theory on the compositionality of idioms for the sake of classifying their sentiment or semantics. Due to the lack of coherence between component-wise sentiment polarities and crowdsourced phrase-level classifications, the findings of their analysis show that idioms are non-compositional for both sentiment and meaning. They conclude that idioms are phenomena in which the non-

compositionality of emotion is not stated or immediately obvious and that the lack of a relationship between component words and phrase-level sentiment necessitates additional research into how to handle idioms in context.

### **3.6 Deep Transfer Learning**

The model parameters of neural networks are initially determined at random before the training process begins, during which they are optimized using various techniques to minimize losses and provide the most accurate results possible. The optimization's primary goal is to adjust the weights to address the vanishing learning rate. At this point, there is a widespread consensus that Deep Learning techniques outperform conventional machine learning approaches about accuracy for the vast majority of NLP tasks [69 - 71]. However, in its infancy, there were several obstacles encountered while using deep learning for natural language processing: To begin, the deep learning models that were available at the time were not sufficiently advanced to run complex models. In the second place, data-driven deep learning models were missing a significant proportion of manually annotated data, which is an activity that is unquestionably laborious and costly. Therefore, researchers gradually start focusing more of their attention on pre-training tactics to find solutions to these issues.

The current undeniable overlords of the NLP are the enormous pre-trained models known as Transformers. Their design aims to handle long-range input and output dependencies with attention and repetition while resolving sequence-to-sequence tasks. Pre-training and self-attention are the two cornerstones of a successful deep learning-based modern NLP system. NLP has found a lot of success using unsupervised representation learning. These strategies often begin



by pre-training neural networks on massive unlabeled text corpora, then tweaking the models with further training on downstream tasks.

A transformer can generate reliable contextualized word and phrase vectors while also keeping track of the whole input sequence thanks to a mix of self-attention methods and thorough unsupervised pre-training. In addition to performing better on empirical testing, pre-trained transformer models may be taught much more quickly than architectures built using recurrent or convolutional layers.

In 2017, Google presented Transformers to the general public for the very first time. When they were first developed, most aspects of natural language processing were taken care of by recurrent neural networks (RNN) and CNN. Even though RNNs and CNNs are both capable of producing accurate results, the Transformer is considered to be a significant improvement over both of these models because it does not require data sequences to be processed in a certain order. This makes it possible for the Transformer to produce more accurate results. Because transformers can process input in any order, they make it possible to train on far larger data sets than was previously possible. As a direct consequence of this, it became much simpler than before to create pre-trained models such as BERT, which, before its release, was trained using massive amounts of linguistic data.

For classification, the model is trained in an environment where annotated data is readily available or simple to collect. After then, it is “fine-tuned” and tested in a field where obtaining training data is challenging. Transformers stand out from other AI systems because they can easily be modified (fine-tuned) to

operate admirably even when learning with little or no data. When utilized off-the-shelf, the majority of them are still useful since they have been heavily optimized and trained on a lot of data. To get the intended output or improve performance on the downstream task, fine-tuning in deep learning includes leveraging weights of a prior model for training another comparable deep learning process.

Although there is no question about the efficiency of transformers' strategies, there are very few formal comparisons and controlled sandbox studies because of a variety of factors [71]. As an example, knowledge bases, ontologies, grammatical characteristics, reasoning, and databases are just a few of the technologies that are frequently used when employing transformer models. As a result, it is challenging to compare and contrast a single pre-trained transformer with alternative approaches. The history of NLP has also seen significant investment in competitions and cooperative projects, where several teams are challenged to use unlabelled datasets to find answers to particular challenges. Even though this has been crucial for the development of NLP research, Lin *et al.* (2021) point out that even minor changes to the initial random seed can have a big effect on model comparison [165].

### **3.6.1 BERT Transformer**

Bidirectional semi-supervised model BERT was pre-trained using unlabeled data from the English Wikipedia and Books Corpus. Over the course of its development, BERT has established new standards in 11 distinct tasks relevant to the comprehension of natural languages, such as sentiment analysis, semantic-role labeling, sentence classification, and the disambiguation of polysemous words. When it comes to handling context and polysemous words,

earlier language models like word2vec had flaws and limits. It's great that BERT was able to overcome these kinds of challenges. Experts in the field of study concur that ambiguity is the greatest challenge to accurately comprehending natural language and that BERT effectively circumvents this barrier. Its language processing skills are on par with those of a human, and it can decipher “common sense” statements.

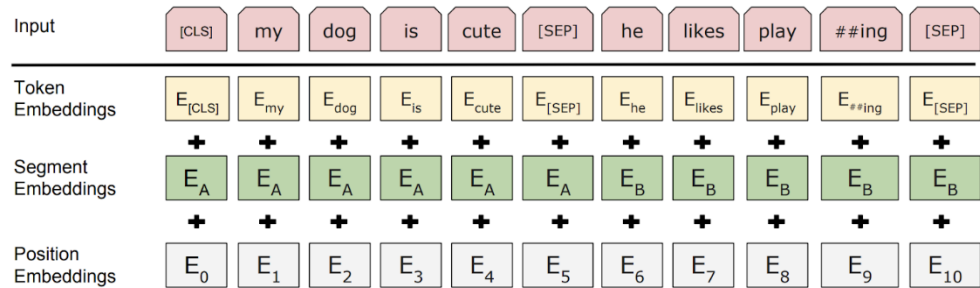
In the 2019 announcement, Google said that they will begin implementing BERT into their production search algorithms in the United States. The percentage of BERT's effect on Google search queries is estimated at 10%. Because BERT strives to provide a wonderful “search experience,” optimizing material for it is not recommended. It is advised that users tailor their content and questions to the context of their typical usage. Through the end of 2019, BERT has been implemented in over 70 different languages. However, researchers can utilize their data to optimize (i.e., fine-tune) these models for downstream tasks (such as classification, entity identification, question answering, etc.) to provide cutting-edge predictions, or they can use them to extract excellent linguistic features from text data. The BERT architecture may be used for several downstream tasks, including named entity recognition, classification, and question-answering. Pre-trained BERTs have earned the moniker “black box” because of their ability to produce  $H = 768$  shaped vectors for input words in a sentence. The sequence may consist of a single or pair of sentences with a separator [SEP] between them and start with a token [CLS] [166]. SEP stands for separator and is used to indicate the boundary between two sentences in a sequence. For example, if we want to analyze the sentiment of a sentence that contains two independent clauses, we can use the [SEP] token to

separate them so that the model can understand that there are two distinct sentences. CLS stands for classification and is used to indicate the start of the input sequence. The [CLS] token is used in the pre-training phase of the model to learn a representation of the entire input sequence. It is also used in the fine-tuning phase for tasks such as text classification, where the final hidden state of the [CLS] token is used as the representation of the input sequence for classification. Therefore, when using pre-trained BERTs, the input sequence of a sentence may consist of one or more sentences with a [SEP] token separating them and start with a [CLS] token. These tokens allow the model to understand the structure of the input sequence and perform tasks such as text classification or sentiment analysis. This means that throughout the training phase, BERT is picking up information from both the left and right sides of a token's context (small units of the surrounding text). BERT predicts disguised words by looking at the words that come before and after a given phrase of words. Several pre-trained transformers motivated by BERT have been proposed, including Roberta, ALBERT, and DistilBERT [43, 44, 158].

### **3.6.1.1 BERT Input Representation**

By employing the unique token [SEP], BERT can distinguish between inputs of one or two sentences. For classification tasks, the text always starts with the [CLS] token. As a result of the nature of the model, the input representations have to be capable of clearly representing either a single text sentence or a pair of text sentences in the same token sequence. Both tokens are always required, even if there is just one phrase and BERT is not being used for classification. The input representation of a particular token is produced by adding the embeddings for that token's related segments and positions, as well

as the token itself. Figure 3.1 provides a graphical illustration of the input representations of BERT tokens.



**Figure 3.1: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings, and the position embeddings (Source: BERT’s paper, Devlin et al., 2018)**

### 3.6.1.2 BERT Embeddings

Word and phrase embedding vectors, for example, may be extracted from text data using BERT. Information retrieval, semantic search, and keyword/search expansion can all benefit from these embeddings. Even when there is no keyword or phrase overlaps, such as when we compare customer searches against previous queries or indexed searches, these representations allow us to correctly return results that fit the customer's intent and contextual meaning. These vectors also have the highly important function of providing future models with high-quality feature inputs. NLP models like LSTMs and CNNs require numerical vector inputs, which frequently necessitates translating linguistic and aural features into numbers.

One-hot encoding or embedding feature vectors may be used to represent words. Words can be then matched against pre-computed fixed-length embeddings created by models like Word2Vec or Fasttext. In contrast to the Word2Vec model, which creates fixed word representations independent of the context in which they occur, BERT develops word representations that are

dynamically aware of the context (other terms surrounding the word). The next two sentences provide an illustration example:

*“The man was accused of robbing a bank.”*

*“The man went fishing by the bank of the river.”*

Unlike Word2Vec, which would generate the same word embedding for “bank” in both sentences, BERT would produce multiple word embeddings for the word “bank” in each sentence (i.e. contextual embedding). Since contextual embeddings capture information beyond blatant disparities like polysemy, they provide more accurate feature representations and improve model performance.

The embedding procedure can be outlined as the following:

- Use the WordPiece embedding rather than the Token Embeddings (the yellow second row). Wu *et al.* (2016) used a lexicon of 30,000 tokens and separated word fragments that were indicated with ##. For example, [tweeting = tweet and ##ing], moreover, the [CLS] unique embedding is always the first token of every sequence [168]. This vector is disregarded for tasks that do not include classification. A single sequence is created by squeezing sentence pairs together and separating them using a specialized token referred to as [SEP].
- For the Segment embedding (third row), if the input is a series of two sentences, a unique learned sentence embedding will be appended to each token of each phrase.

We only use the embeddings from the first sentence for inputs that include a single sentence.

- Regarding the Position embedding, which is denoted by the fourth row: in terms of languages, the order in which each word is placed inside a sentence is very significant; hence, the tokens' positions will be designated as Position embeddings.

The total length of the sequence cannot exceed 512 tokens, which is the most that BERT will allow. Sequences that are longer than 512 tokens, regardless of whether they consist of a single phrase or sentence pairs, will be broken up into smaller parts at intervals of 512 tokens. While taking into account how efficiently computing can be performed, BERT often breaks the sequence into chunks with a length of 128 tokens. In the end, BERT will complete the input representation by combining the aforementioned three kinds of embeddings. And to pre-train the model, BERT will make use of the input representation that was generated before.

When building a model like BERT, the normal practice is to start with “pre-trained weights” and then retrain for a small number of trials on a supervised dataset [9]. Observations of model behavior account for the majority of what is now known about what transpires during this time of fine-tuning. Transformers that have been fine-tuned can perform at the greatest level, but there is a chance that they will also take up prejudices and obvious shortcuts [169-171].

### 3.6.1.3 BERT's Pre-training Process

BooksCorpus (800M words) and English Wikipedia (2,500M words) are combined to form the pre-training corpus for BERT, from which two BERT-Base and BERT-Large are generated [172]. A total of 110M/340M parameters may be calculated from components of BERT-base and BERT-Large respectively as shown in Table 3.3.

### 3.6.2 Transformer Fine-tuning

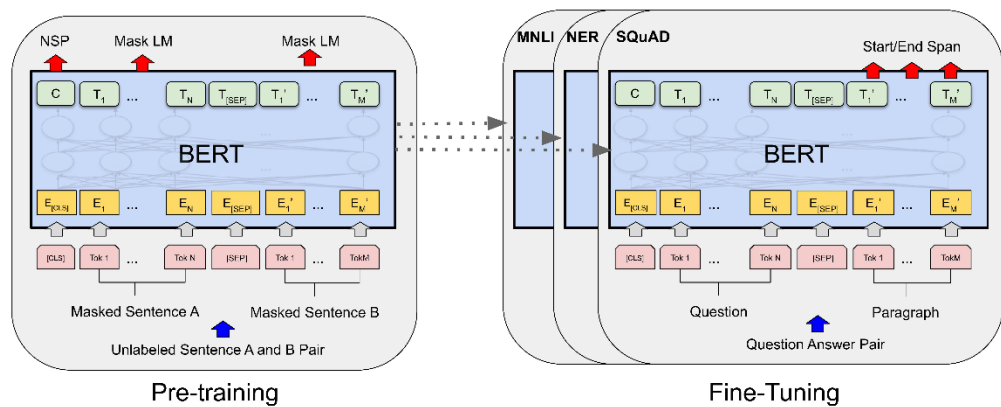
Building pretrained transformers and fine-tuning paradigms have been successfully applied to numerous NLP tasks in recent years, especially in the field of sentiment analysis [68]. Howard and Ruder suggested pre-train a language model on a huge corpus and then tweaking it on the target task [173]. Unique features of this model include slanted triangle learning rates and gradual unfreezing. Researchers are motivated by the impressive outcomes achieved by pre-trained models to achieve exceptional results, even when working with limited amounts of annotated data.. However, increases in the size of training datasets tend to improve the effectiveness of previously trained models [174, 175]. For numerous applications, including language modeling and masked language modeling, pre-trained transformers are commonly deployed.

**Table 3.3: Parameters Setting of BERT-Base and BERT-Large**

<b>Components</b>	<b>BERT-base values</b>	<b>BERT-Large values</b>
L: Number of Layers	12	24
H: Hidden layer size	768	1024
A: self-Attention heads	12	16



Fine-tuning the BERT model produced from a pre-training phase is quite straightforward and may be used for a wide variety of applications. Tasks in Figure 3.2 represent sequence-level classification problems, for which the original pre-trained BERT model can be fine-tuned using different modification settings for different tasks; such as named entity recognition and sentiment analysis. For classification, BERT employs the prediction obtained by feeding the first token's output representation through a softmax classifier.



**Figure 3.2: Overall pre-training and fine-tuning procedures for bi-directional encoder representations from the transformer (BERT)** (Source: BERT’s paper, Devlin et al., 2018)

### 3.7 Data Augmentation for Sentiment Analysis Task

To accomplish comprehensive classification using bag-of-words frequency models, keyword-based sentiment analysis was first created. These approaches suffer from two major flaws: they fail to account for the “sentimental” association of words in the local context and they fail to show the transmitted “hidden meaning.” The problem has been suggested to be solved in several different ways. Even though contextual data augmentation manages the interword correlation, it is only used sparingly in Natural Language Processing (NLP), especially in sentiment analysis tasks [176]. One possible explanation

for this is that no set of criteria exists for converting data between different fields or domains that can be used universally [177].

Data augmentation is something that researchers are especially interested in pursuing in the domains of computer vision and speech recognition. Text data augmentation, on the other hand, has received far less research and does not adhere to any particular methodological norm. Utilizing a dictionary, a thesaurus, or a database of synonyms is the most typical approach to replacing words with their synonyms. When there is no dictionary available to use, one choice is to make use of distributed word representation to locate comparable words. The term “synonym augmentation” refers to the procedure being described here. Although consciously changing the terms of the language would be the best way to augment, this approach would be prohibitively expensive. As a result, the most workable strategy for increasing the amount of data in the majority of research is to increase it by exchanging certain terms or phrases with their equivalents [68]. WordNet is the most popular open-source lexical database for the English language [178]. However, word embedding and distributed word representation are used in the newly developed technique known as semantic similarity augmentation, which helps discover semantically connected phrases. For this method to work, we will either need to have word embedding models that have been pre-trained for the appropriate language or a sufficient quantity of data from the application that we want to use. When searching for synonyms, this removes the need for any other dictionaries to be consulted, which is one of its many benefits. Another approach, which is referred to as reverse translation, involves first translating phrases, paragraphs, or individual words into another language (which is referred to as forward

translation), then translating the results back into the first language [179, 180].

Generally speaking, the most prevalent NLP augmentation approach is to replace synonyms selected from a manual taxonomy [181]. Word similarity is used by [182] to augment data. In addition, Nicolai *et al.* (2022) use many phoneme-font translation strategies [183]. In contrast, the word-to-word synonym substitution augmentation approach disregards the strong polarity comparison of emotional terms while doing sentiment analysis [15]. Other researchers presented an augmentation approach that generates fresh data by translating sentences from one language to another [184, 185]. Previous research [129, 186] has also made use of data noise as smoothing and predictive language models for synonym replacement. Even though these tactics are successful, they are seldom adopted owing to the high implementation costs associated with the performance benefits they provide. Rizos *et al.* (2019) developed and analyzed three augmentation procedures to reduce class imbalance and maximize data collection from scarce sources; these are known as substitution-based augmentation, word position augmentation, and neural generative augmentation [187]. Another common method for improving lexical data is to use a thesaurus or ontology to replace words. Zhang *et al.* proposed a method utilizing character-level CNN to swap out lexical parts [65]. They used two geometric distributions to determine how many words needed changing and then used a ranked list of choices to choose the best one.

Another word embedding-based augmentation method employed the cosine similarity between words and framed word representations as its measure. With this method, we can locate a viable replacement for the target

terms by looking at their k-nearest neighbors [182]. Even though they lacked knowledge of grammatical rules, they performed better on a task that included topic classification. In previous research on Twitter posture detection, Word2Vec was used to discover candidates ranked by the cosine similarity between Word2Vec vectors [32, 188]. The BERT model was used by Souza & Souza (2021) to analyze the sentiments conveyed in Brazilian and Portuguese product evaluations by the utilization of word embedding [189]. They believe that the BERT fine-tuning made it possible for the model to perform better than before.

The fundamental problem with word embeddings is that they frequently combine many word meanings into one confused vector. According to Yagoobzadeh *et al.* (2019), conventional word embedding techniques “learn embeddings that capture numerous meanings in a single vector well – assuming the meanings are frequent enough,” [190]. They discover that difficult cases of ambiguity, such as words with many meanings or unusual word meanings, are better represented when the dimensionality of the embedding space is raised. Şahin (2022) released a relatively recent paper on the advantages of augmentation for NLP tasks. According to Şahin, part-of-speech tagging, semantic role labeling, and augmentation considerably enhance dependency parsing [191].

### **3.8 Summary**

A powerful method for automatically analyzing unstructured data is known as sentiment analysis, and it is often used in conjunction with text analysis. The aim of this research area is to examine user-generated content on social media platforms such as Twitter, with the goal of obtaining insights into users' motives,

emotions, and worldviews. It is one of the most commonly researched issues in natural language processing, and the disciplines of data mining, web mining, and social media analytics have all devoted a large amount of resources to the study of sentiment analysis.

In this Chapter, we have discussed the earlier approaches that were made to tackle the sentiment analysis task, including the lexicon-based, machine learning, and deep-learning techniques, along with their respective benefits and drawbacks. In addition to this, the Chapter delves into the many augmentation strategies that are often used while performing the work of sentiment analysis.

## CHAPTER 4

### RESEARCH METHODOLOGY

#### 4.1 Introduction

In this chapter, we describe the proposed research methodology for enhancing sentiment analysis of tweets containing idiomatic expressions. The methodology consists of the following tasks:

##### 1. **Data Collection:**

The first task in this study is to collect the Twitter data that will be used for sentiment analysis. We will use the Twitter developer API to obtain a dataset of English tweets related to various topics, such as politics, sports, and entertainment. We will focus on tweets that contain idiomatic expressions and use them as the basis for our analysis.

##### 2. **Creation and Compilation of Idioms List Using External Online Resources:**

Next, we will create a list of idiomatic expressions by crawling online dictionaries and thesauruses, such as Merriam-Webster and Roget's Thesaurus. We will also use online idiom lexicons, such as The Free Dictionary and Idioms Online, to compile a comprehensive list of idiomatic expressions.

##### 3. **Crowdsourcing Service to Manually Annotate Idioms with Their Sentiment Polarity:**

We will then use a crowdsourcing service, such as Amazon Mechanical Turk, to manually annotate each idiomatic expression in our list with its sentiment

polarity. The annotators will classify each idiom as positive, negative, or neutral based on their understanding of the idiom's meaning and context.

**4. Compilation of a Gold Standard Lexicon by Merging the Produced Lexicon with the SliDE Lexicon Available at IBM Website:**

We will merge the annotated idioms with a pre-existing sentiment lexicon of idiomatic expressions, such as the SliDE lexicon available at IBM's website, to create a gold standard lexicon for sentiment analysis of idiomatic expressions.

**5. Extraction of Opinionated Tweets for Each Idiomatic Expression in the Lexicon Using Twitter Developer API:**

Using the Twitter developer API, we will extract a set of tweets for each idiomatic expression in the gold standard lexicon. We will focus on tweets that contain the idiomatic expression and have a clear sentiment polarity.

**6. Expansion of Idioms by Crawling Online Thesaurus and Dictionaries to Retrieve Their Formal Definitions:**

To enhance the contextual understanding of idiomatic expressions, we will expand our list of idioms by crawling online dictionaries and thesauruses to retrieve their formal definitions. This will help us better understand the meaning and context of each idiom and improve the accuracy of sentiment analysis.

**7. Selection and Fine-tuning the BERT-variant Transformer:**

We will use a BERT-variant transformer, such as RoBERTa or DistilBERT, as our sentiment classifier. We will fine-tune the transformer using the annotated tweets and the gold standard lexicon to improve its ability to identify the sentiment polarity of idiomatic expressions.

**8. Identification of the Polarity Expressed in the Tweets/Idiomatic Expressions Using the Transformer, and Classify Them into Tweets as Positive, Negative and Neutral:**

Using the fine-tuned transformer, we will classify each tweet in our dataset as positive, negative, or neutral based on the sentiment polarity expressed in the tweet. We will also classify each idiomatic expression in our lexicon using the transformer to identify its sentiment polarity.

**9. Evaluation and Comparison of the Results of the Expansion-Based Classification with the Gold Standard Counterparts:**

Finally, we will evaluate the performance of our expansion-based classification approach by comparing its results with those of the gold standard lexicon. We will use metrics such as precision, recall, and F1-score to assess the accuracy of our approach and identify areas for improvement.

**Recap of Research Questions:**

These are the five research questions that will guide our work on enhancing sentiment analysis of tweets containing idiomatic expressions:

RQ1: How can we efficiently build and annotate a sentiment lexicon of idiomatic expressions using external knowledge bases?

This research question aims to explore the most efficient ways to leverage external knowledge bases to build a sentiment lexicon of idiomatic expressions. This includes identifying relevant sources of information and methods for annotating and categorizing idiomatic expressions based on their sentiment.

RQ2: What is the impact of incorporating idiomatic expressions as features on the sentiment classification of tweets?



This research question seeks to determine the extent to which incorporating idiomatic expressions as features can improve the accuracy of sentiment classification of tweets. It includes exploring the most effective ways of representing idiomatic expressions in feature vectors and analyzing the impact of different feature selection techniques.

RQ3: How does leveraging external knowledge bases enhance the performance of sentiment analysis of tweets containing idiomatic expressions?

This research question aims to investigate the effectiveness of leveraging external knowledge bases in enhancing the performance of sentiment analysis of tweets containing idiomatic expressions. It includes analyzing the impact of different external knowledge bases and identifying the most effective ways of incorporating external knowledge into sentiment analysis algorithms.

RQ4: How to perform a sentiment classification of tweets with idiomatic expressions while having little or no training data?

This research question aims to explore how to perform sentiment classification of tweets containing idiomatic expressions when there is little or no training data available. This includes exploring Deep learning approaches (mainly pre-trained transformers) to sentiment analysis and identifying the most effective methods for leveraging external knowledge in these approaches.

RQ5: To what extent does the use of data augmentation and normalization pre-processing procedures influence the accuracy of the sentiment classifier?

This research question seeks to determine the impact of data augmentation and normalization pre-processing procedures on the accuracy of the sentiment classifier. It includes exploring different data augmentation and normalization techniques and identifying the most effective ways of incorporating these techniques into the sentiment analysis pipeline.

The proposed methodology is designed to address these research questions by using a variety of techniques and tasks, such as data collection, crowdsourcing, lexicon compilation, transformer fine-tuning, and evaluation.

## **4.2 Methods Selection Criteria**

Osgood *et al.* (1957) hypothesized that the semantic orientation of words could be quantified [192]. Based on this assumption, several works on sentiment analysis have presented techniques for classifying sentiments as positive, neutral, or negative polarities. Frequently, the classification methods may be divided into two primary categories: machine learning approaches [193] and semantic orientation approaches. The latter entails generating sentiment lexicons using dictionaries such as WordNet or other statistical approaches such as word co-occurrence approaches. Using bootstrapping techniques, lexicon-based systems generate opinion word lexicons from a short list of opinion words, their polarities, and their synonyms. Machine learning methods including supervised, unsupervised, and deep learning dominate sentiment analysis. However, their success relies on the quality and quantity of the training data. Some researchers such as Medhat *et al.* claimed that a hybrid method of the main approaches may also be used in sentiment analysis [194].

By investigating the existing methods of sentiment classification and the problem of idiomatic expressions used in tweet data, selecting the proper

method is very crucial. Therefore, the scope of the thesis and the considerations to solve the existing problem are set as follows:

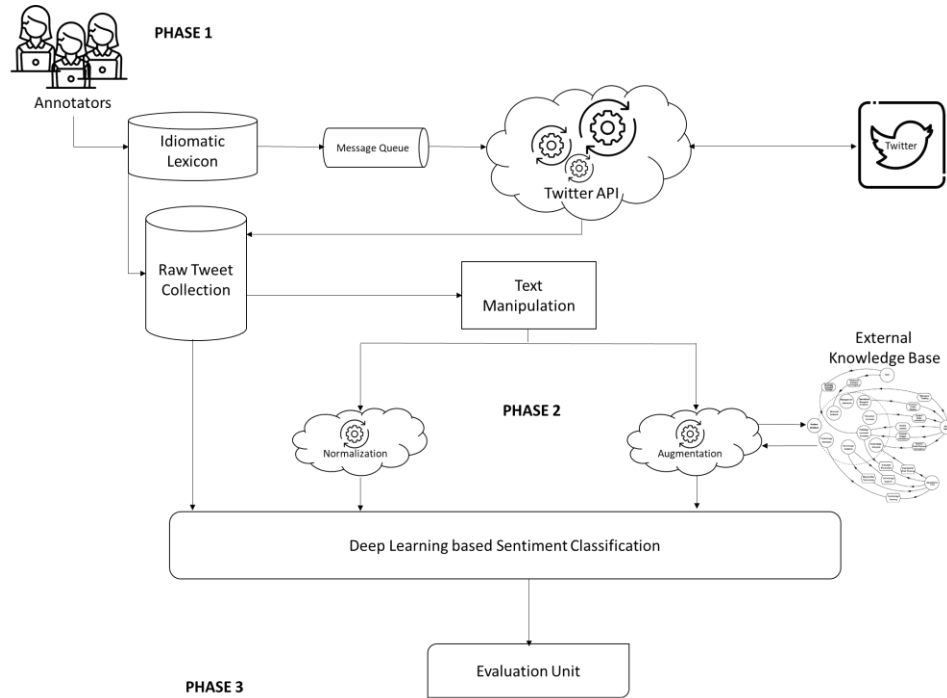
- 1- An idiomatic expression is rich with semantic orientation and conceals sentiment polarity. However, to manually collect and annotate existing and ever-updating idioms or another figurative language to build a reference lexicon for sentiment classification is impractical. It's also very hard if not useless to collect instances from all idiomatic expressions and label them to train a supervised model for sentiment classification. In addition, although we can avoid training by unsupervised learning, however, the literal construction of the idiomatic expressions does not reveal the actual sense or meaning behind the idiom and therefore it's not possible to utilize unsupervised learning to solve the problem at hand.
- 2- Transfer learning is a very powerful paradigm that can be used to avoid training models from scratch. However, usually transformers are built to solve a specific NLP task or trained on generic documents. Therefore, to utilize transfer-based learning, one should fine-tune or retrain transformers to handle a new downstream task. Although fine-tuning and retraining are much easier than building or training a transformer from scratch, this process can lead to unstable results in the classification performance because every time we perform the retraining process, it starts with random initial parameters which might lead to fluctuating accuracy.
- 3- Applying traditional data augmentation methods such as word-to-

synonym substitution and backward translations cannot reveal the actual sentiment of idiomatic expressions and therefore there should be a method to reflect the sentiment of the overall “actual meaning” of the idiomatic expressions.

- 4- Building an idiom recognition method to extract idiomatic expressions within a text is useful, especially for real-time idiom extraction. However, this process can be relaxed by a simple string comparison between the idiomatic lexicon entries and the input text because most idioms expose a static structure. The ultimate goal of this thesis is to utilize idiomatic expressions in the sentiment classification of tweet data. Thus, building and enhancing the idiom recognition tool is out of the scope of this thesis and therefore can be further investigated in future work.
- 5- The positional context of words within sentences can change the overall meaning of the sentence. Therefore, the bag-of-words method always fails to keep the structural distribution of a given sentence and therefore it's important to keep the positional order of words while building the feature vector. Deep learning models such as LSTM can hold the context however for shorter sentences and can't handle long sentences. The state of art to solve this problem is the self-attention mechanism that is used as the basis for transfer-based learning.

Throughout the careful selection of the methods and design process, we came up with a suggested framework structure that was created modularly with

the research guidelines, scope, and objectives in mind. As shown in Figure 4.1, the first step aims to gather and assemble a list of idiomatic phrases and annotate them using a crowdsourcing service. This step allows us to construct a reference lexicon to verify and assess our idiom expansion and annotation method.



**Figure 4.1: The Proposed Framework**

Since our ultimate goal is to classify sentiments of idiomatic tweets (tweets with idioms), we used the Twitter API to connect and retrieve tweets by formulating idiomatic queries from the lexicon. In the idiom expansion step, we connect to an external knowledge base to recall definitions and meanings of idioms. In addition, we compare the expansion method to other common data augmentation methods used in sentiment analysis. Throughout the experimentations, we've noticed that some text preprocessing alters and influences the sentiment classifier's performance. The details are discussed in the experiments section.

BERT Transformer has different versions with different goals to achieve.

We use roBERTa for the following reasons: 1) roBERTa was pretrained for a sentiment classification task, 2) roBERTa classifies the input into three sentiment classes (positive, negative, and neutral), 3) roBERTa output includes the calculated percentages (probability score) of the sentiment classes over the input text.

### 4.3 Datasets Preparation

#### 4.3.1 Idiomatic Expressions Preparation

Besides the idiomatic expressions offered by the SliDE lexicon, we extract idioms randomly from “The Free Dictionary,” “Education First,” and the “Oxford Dictionary of Idioms.” We manually filter out the sentiment-bearing idioms, and the final list contains 3,930 idiomatic expressions different from those found in SliDE (the IBM Sentiment Lexicon of Idiomatic Expressions contains 5,000 idiomatic expressions).

A sample of the compiled list of idioms is shown in Table 4.1. Initially, the list of idioms consists of 6,400 idioms and there is no label assigned to them. After filtering the idiom list, we kept 3930 idioms. To keep the search more flexible, we parenthesized the optional letters or keywords that can be changed while using the idiom in the text.

**Table 4.1: Sample of the collected idioms from online dictionaries and thesauruses**

<b>Idiom</b>	<b>Positive</b>	<b>Neutral</b>	<b>Negative</b>
(A) bigger bang for your buck	0	0	0

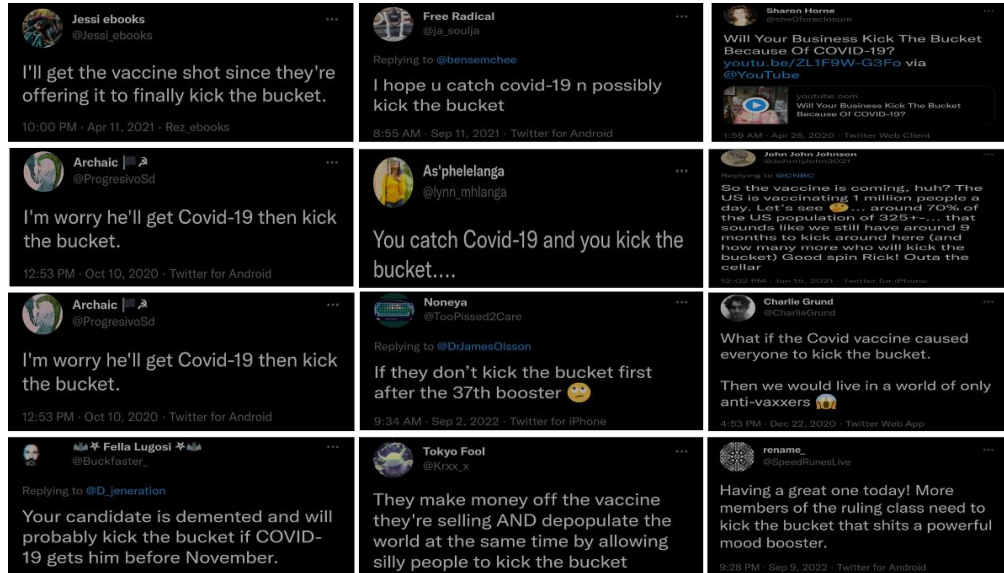
(A) Dog is a man's best friend	0	0	0
(A) watched pot never boils	0	0	0
(An) eye for an eye, a tooth for a tooth	0	0	0
(As) different as chalk and cheese	0	0	0
(As) thick as thieves	0	0	0
(As) thick as two short planks	0	0	0
(Ask not) for whom the bell tolls	0	0	0
(Beware the) Ides of March	0	0	0
(By the) skin of your teeth	0	0	0
(Can't) hold a candle to	0	0	0
(Coming in) on a wing and a prayer	0	0	0
(Go to) Hell in a handbasket	0	0	0

### 4.3.2 Tweet Data Collection

Twitter users share anything from personal updates and snippets of their day to news stories, songs, and essays in which they voice their views and discuss current events. Twitter's utilization as a forum for self-disclosure makes it a great resource for opinion-charged discussions platform [148]. In light of this fact, it has lately emerged as a primary resource for textual information for several NLP tasks including sentiment analysis [195]. We use the developer API to retrieve and collect tweets by formulating idiomatic queries. We aim to retrieve “tweets with idioms” rather than searching for idioms by implementing linguistic pattern-matching (such as string matching or regular expressions) to detect whether a tweet has an idiom expression. In this case, each query returns a dynamic number ( $\sigma$ ) of desired tweets. For the sake of experimenting, we set  $\sigma = 50$  and retrieved  $50 * 8930 = 446,500$  tweets using the implementation of the “Tweet Database Creation” algorithm as shown in Table 4.2. Figure 4.2 shows a sample of the retrieved tweets using the idiomatic query “kick the bucket”.

It is necessary to get the Twitter API credentials, which include the key and secret passwords, to utilize the Twitter Application Programming Interface

(API). Using the API, query parameters about certain keywords may be defined. These parameters include “search by,” “language,” “allow retweets,” and so on. The data that is collected can be stored in Comma-Separated Value (CSV) format.



**Figure 4.2: Sample of Retrieved Tweets using Idiomatic Expressions in the API Query**

**Table 4.2: Pseudo Code of Tweet Collection Algorithm**

Algorithm 1 Tweet DB Creation
Output $\leftarrow \mathcal{D}$ : Final Tweets Dataset
Input $\leftarrow \mathcal{F}$ : Lexicon of Idiomatic Lexicon; $\sigma$ :max # tweets per idiom
Step 1: Initialization
$\mathcal{F}_{temp} \leftarrow \phi$ ; $\mathcal{D} \leftarrow \phi$ ;
Step 2: Iterate over the idioms & retrieve relevant tweets
<b>for all</b> $i \in \mathcal{F}$ <b>do</b>
count $\leftarrow 0$ ;



```

While  $count < \sigma$ 

     $\mathcal{F}_{temp} \leftarrow \mathcal{F}_{temp} \cup \text{API.Query}(i)_{fulltext}$ 

     $count ++$ ;

end while

 $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{F}_{temp}$ ;

end for

Output :=

 $\mathcal{D} \leftarrow$  Tweet collection containing idiomatic expressions

end procedure

```

## 4.4 Data Annotation

### 4.4.1 Idiomatic Lexicon Annotation

Similar to the assumption used to build SLiDE, we create and annotate the sentiment lexicon using a crowdsourcing service of ten annotators. Each idiom is annotated by a positive, negative, or neutral label based on the highest number of votes it receives. In total, the new lexicon contains 8,930 distinct idiomatic expressions. If an idiom receives similar votes among the sentiment classes, we use the following priority order to resolve the conflict: Negative > Positive > Neutral. The definitive lexicon contains 2,612 positive labels, 2,892 negative labels, and 3,426 neutral, as shown in the distribution table 4.3.

**Table 4.3: eSLiDE Idioms' Polarity Distribution**

<b>Polarity of idioms</b>	<b># of idioms</b>
Positive	2,612
Negative	2,892
Neutral	3,426

Although the final lexicon has 12 columns as used in SLiDE, we drop

columns 6 and 12 as they are not relevant to our research focus. Table 4.4 describes the content of each column and Table 4.5 shows an example of the lexicon entry contents.

**Table 4.4: The SLiDE Lexicon Columns Structure**

<b>Column number</b>	<b>Column description</b>
1	The idiomatic Expressions
2	The online web reference of the idiom
3	number of positive annotations
4	number of negative annotations
5	number of neutral annotations
6	<i>Number of profane or inappropriate annotations</i>
7	Total annotations received
8	Positive annotation percentage
9	Negative annotation percentage
10	Neutral annotation percentage
11	Majority voting label
12	<i>Ambiguous expression filter</i>

After mapping all annotations into sentiment scores, we assess the training dataset's reliability by calculating inter-annotator-agreement (IAA) using Krippendorff's alpha [196]. This measure was chosen as a generalization of established reliability indices due to the following properties: (1) it can be used with more than two annotators, (2) it can be used with an arbitrary number of categories, (3) it may be utilized with incomplete or absent data, (4) it accounts for changes in the expected or predicted IAA [197]. According to equation (4.1), Krippendorff's alpha coefficient is computed as 4.1:

$$\alpha = 1 - \left( \frac{D_o}{D_e} \right) \quad (4.1)$$

The proportion of annotations on which the two annotators disagree, computer by  $D_o$ , indicates the true degree of disagreement. When annotating data at random,  $D_e$  reflects the typical discordance that may arise. For data to be regarded as a reliable training set, Krippendorff suggests a threshold of = 0.667. A web-based inter-annotator agreement tool was used to calculate values for Krippendorff's alpha [198]. On the idiomatic dataset, the agreement was estimated as  $\alpha = 0.696\%$ , where  $De = 0.701$  and  $Do = 0.213$ .

We conduct two distinct experiments to manually annotate the lexicon. The first annotation was done by a paid crowdsourcing service. In this service, English natives were asked to annotate the idioms provided in excel format. We provided the annotators with an empty excel sheet and they select from a drop-down box the proper tag for each idiom. In the second annotation, we asked ten volunteers (university students who are non-native English speakers) from different faculties to annotate the idiomatic expressions using a simple form-like tool to tag idioms as shown in Figure 4.3. The form consists of three parts: 1) the first part shows the idiomatic expression to be annotated and its definition or meaning (we expect that annotators of this experiment are unfamiliar with the actual meanings of idioms), 2) the second section of the form asks the annotators to select the sentiment tag/label they think it's more appropriate for the idiom, 3) the last section asks about the confidence level that the annotator feels when selecting that tag.

In the end, we compare the annotation results of the two groups. We find that providing a definition/meaning of an idiom was very fruitful and the "final tag" similarity among the groups was 97%. However, the actual voting

percentage was different for the majority of the idioms. For example, the idiom “kick the bucket” received (0 positives, 5 negatives, and 5 neutrals) votes from the crowdsourcing service. On the other hand, the second group votes for the same idiom as (0 positives, 10 negatives, and 0 neutrals) votes. Although the voting was different, the final assigned tag was “NEGATIVE” for both cases. The above result gives us a clue that we might utilize the free-of-charge student group to annotate the tweets collection to build the baseline test dataset.

The image shows a mobile application interface for sentiment annotation. At the top, it displays the user's email 'bashar.tahayna@1utar.my' and a 'Switch account' link. Below this is a tweet snippet: '(1) bigger bang for your buck' with a description: 'Greater value for the amount of money one is spending'. The main part of the interface is a form with two sections. The first section is titled 'Proper label \*' and contains four radio button options: 'Positive', 'Negative', 'Neutral', and 'Unknown'. The second section is titled 'Your confidence level: \*' and contains three radio button options: 'Low', 'Medium', and 'High'. At the bottom of the form, there are three buttons: 'Back', 'Next', and 'Clear form'.

**Figure 4.3: Annotation platform interface**

#### **4.4.2 Gold Standard Tweets Annotation**

To create a labeled benchmark dataset of tweets containing idiomatic phrases, we automatically annotate tweets based on the labels of their corresponding idiom they contain. This assumption cannot hold all the time as

idioms could refer to different sentiments based on the context they appear in. For example, “break ranks” has received a “negative” label while it can be “neutral” or even “positive” in some contexts. For example, the tweet “*Henry Sy Jr break the ranks and became richer than Elon Musk*” can be classified as a positive or neutral sentiment, while the “*This legislation is tailored for anyone who breaks the ranks in this country*” tweet has received a negative sentiment. In Table 4.5, some idioms might have different sentiment than it was labeled by the voting.

**Table 4.5: Sample Content of the SLIDE Lexicon**

Idiom	Pos	Neg	Neu	Inapprop.	Total	%Pos	%Neg	%Neu	Maj. Label
back to the wall	0	5	5	0	10	0	0.5	0.5	negative
beg to differ	0	5	5	0	10	0	0.5	0.5	negative
best thing since sliced bread	5	0	5	0	10	0.5	0	0.5	positive
beyond the pale	0	5	5	0	10	0	0.5	0.5	negative
big picture	5	0	5	0	10	0.5	0	0.5	positive
big up	5	0	5	0	10	0.5	0	0.5	positive
blot out	0	5	5	0	10	0	0.5	0.5	negative
bottom out	0	5	5	0	10	0	0.5	0.5	negative
break ranks	0	5	5	0	10	0	0.5	0.5	negative
bridge the gap	5	0	5	0	10	0.5	0	0.5	positive
bright line	5	0	5	0	10	0.5	0	0.5	positive
bright-line rule	5	0	5	0	10	0.5	0	0.5	positive
bring to the table	5	0	5	0	10	0.5	0	0.5	positive
brush up	5	0	5	0	10	0.5	0	0.5	positive
buckle down	5	0	5	0	10	0.5	0	0.5	positive

#### 4.5 Data Augmentation Methods

The term “data augmentation” describes techniques used to expand the amount of data by adding copies of current data that have been significantly updated or by generating brand-new synthetic data from existing data. Improving the variety of training data is one of the key goals of data augmentation techniques since doing so will aid the model’s ability to generalize to new testing data. Augmenting text data in NLP is a modern method compared to image data augmentation in Computer Vision. Images may be easily transformed by simple processes like flipping them or turning them in grayscale without losing any of their meaning. Since augmentation is semantically

invariant, it has become a crucial technique in the field of Computer Vision.

To increase model generalization on downstream tasks, supplemented data is also anticipated to be different based on validity. The diversity of enhanced data is included here. According to the variety of their enhanced data, we may classify data augmentation techniques into three groups: paraphrasing, noising, and sampling.

#### **54.5.1 Paraphrasing-based approaches**

Based on appropriate and constrained alterations to phrases, these approaches produce enhanced data with little meaning different from the original data. The enhanced data communicate information that is substantially close to that in the original form.

##### **4.5.1.1 Thesaurus**

Using a thesaurus, we substitute a word at random from the phrase with its synonym in this method. For instance, we search for the synonyms in the WordNet and ConceptNet databases before replacing them. Although this method can improve the text data by adding synonyms, it can corrupt the actual meaning if applied to idioms.

##### **4.5.1.2 Semantic Embeddings**

Semantic embeddings are a type of representation of words or phrases in a high-dimensional vector space, where the position of each vector corresponds to the meaning of the represented word or phrase. These embeddings are created using a deep learning algorithm, typically a neural network, which learns to map each word or phrase to its corresponding vector in the high-dimensional space.

The key feature of semantic embeddings is that they capture the semantic relationships between words, such that words with similar meanings are located close to each other in the vector space. For example, the embeddings for the words "cat" and "dog" would be located closer to each other than to the embeddings for the words "car" or "tree", as the former two words are more semantically similar to each other than to the latter two.

This strategy eliminates the need for a thesaurus, which is limited in its ability to substitute idiomatic expressions and phrases. Using previously taught word embeddings, the original word in the sentence is swapped out for its closest neighbor in the embedding space. Here, we can make advantage of pre-trained word embedding to replace a word in a phrase with its closest neighbor terms from the embedding space. Because of this, we'd be able to portray the intended meaning of the text [51].

#### **4.5.1.3 Masked Language Model**

To train BERT and other transformer models, a large quantity of text was used in conjunction with a pretext task referred to as “Masked Language Modeling” (MLM). This task challenges the model to predict masked words based on the surrounding context. This model can be utilized to enhance tweets by masking sections of text at random and then asking the model to guess which token corresponds to the masked content.

#### **4.5.1.4 Machine Translation**

This method relies on machine translation to retrain the meaning of a document by paraphrasing it. We follow the procedure of back-translation as suggested in:

1. Convert a statement from the English language to another language.
2. Reverse-translate the phrase into English.
3. Verify that the new statement differs from the one we started with.  
If so, we utilize this new phrase as an improved version of the first one.

#### **4.5.2 Noising-based Approaches**

Through their comprehension of linguistic phenomena and past information, humans considerably limit the influence of faint noise on semantic understanding, but this noise can provide problems for models. Thus, this approach increases model resilience while increasing the training data.

##### **4.5.2.1 Swapping**

Swapping can be done at the word or sentence level. We apply this method at the word level to replace words' positions in the same sentence as shown in the example below.

*It is an awesome hotel. → SWAP WORDS → It is a hotel awesome.*

##### **4.5.2.2 Deletion**

The extreme issue with this method is when an idiom or slang consists of one word as in the idiom “catch-22”. We drop the idiom or lose part of its constituting words in such cases.

##### **4.5.2.3 Insertion**

This method's implementation is straightforward. Here, we randomly choose



a term from a tweet or idiom, replace it with a synonym, and then re-introduce the synonym term into the idiom/tweet. When we apply this method one time to the idiom “hit the hay”, which means go to sleep, the word “hay” was selected randomly, and insert back its random synonym “straw” at a random position. We got “hit straw the”, which might make no sense, after the insertion process.

#### **4.5.2.4 Substitution**

With this technique, random strings are used in place of words or phrases. This paraphrase technique often stays away from employing strings that are semantically comparable to the original material, in contrast to the approaches discussed above.

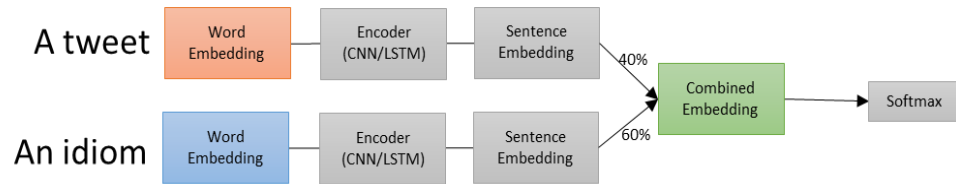
### **4.5.3 Sampling-based Approaches**

This method can sample novel data within the data distributions that they have mastered. It produces a wider variety of data and meets more downstream task requirements based on trained models and artificial heuristics.

#### **4.5.3.1 SentMixup**

Instead of generating text in a natural language format, this method enhances samples using simulated embeddings. Because it is based on the existing data, the sampled data in the virtual vector space may have different labels than the original data. This method entails adding zeros to both a tweet and an idiomatic expression such that they both measure exactly  $n$  characters. Using the LSTM/CNN encoder as shown in Figure 4.4, we take the hidden state

after each word is embedded and transform it into a sentence embedding. These embeddings are combined with others in a certain percentage before being sent to the final classification layer. The cross-entropy loss is calculated using both labels of the initial phrases in the given ratio.



**Figure 4.4: Sentmixup Approach Combines Existing Samples by Averaging Sentence Embeddings**

#### 4.6 Data Expansion

The freshly produced tweets must have a comparable tone to the originals. Therefore, we have to ensure that the learning model will be exposed to idioms' meanings rather than merely replacing words with their corresponding counterparts. Because various augmentation approaches may result in different sentiments or provide no utility in determining the sentiment, there are certain circumstances when we cannot guarantee preserving the same polarity of the tweet. The phrase "I had a blast" may signify either "I had a great time" or "I lost my mind." That's why the model needs to be able to understand, with the help of the context, what the phrase means. To do this, we apply a twitter enrichment/expansion technique to accommodate all alternative interpretations of the tweet.

To simplify the recognition of the idiom used in the tweet, we added a new column in the tweet collection with the original idiom as a marker for the idiom used. Although the proposed framework was designed to perform offline idiom expansion, the framework is expandable to can work on the fly without

the need to prepare the idiom expansion versions in the database. Therefore, we modify the idiom expansion algorithm to work in a real-time fashion by adding a pattern-matching grammar procedure to handle the detection of the idioms within a given tweet/text. Most idioms can be easily extracted due to their rigid forms. It is possible to build a basic string matching and lookup function; but, to make this framework more amenable to improvements, it would be preferable to save and retrieve any syntactic modifications that may have been made. For this reason, we used the “My Information eXtraction and Understanding Package” (Mixup) to create the criteria for recognizing lexico-syntactic patterns (Cohen, 2004). One such pattern-matching language is RegEx, as shown by the following example:

*⟨idiom⟩ : ⟨PRP\$⟩ ⟨AUX\$⟩ a blast*

*⟨VB⟩ : had / have / has*

*⟨PRP\$⟩ : we, he, she, they, it, i*

The tweets “@MelanieLDeal76: I had a blast at trump rally last night. 5:17 AM · Jun 22, 2020” and “@erinpauken: We were at the game Sat night. The Pittsburgh fans were fantastic. We had a blast with them. 7:14 AM · Sep 13, 2022” are good examples of where we effectively identify the expression “someone had a blast.” The technique shown in Table 4.6 involves swapping out an idiom with its “definition” from external resources through the DOM function. If the idiom cannot be located in Oxford Dictionary and the query returns null, we go on to another resource. In this example, the tweet “Fauci is a political **yes man**” is mapped to “Fauci is a politically weak person who always agrees with their political leader or they're superior at work,” which

describes the meaning of the yes-man idiom.

**Table 4.6: Pseudo Code of the Idiom Expansion Procedure**

<b>Algorithm 2</b> IdiomExpansion
Output $\leftarrow \mathcal{F}_{temp}$ : Expanded Idioms
Input $\leftarrow \mathcal{F}$ : Lexicon of Idiomatic Lexicon; $\alpha$ : External KB; $\mathcal{D}$ : tweets dataset
Step 1: Initialization
$\mathcal{F}_{temp} \leftarrow \phi$ ; $x_{temp} \leftarrow \phi$ ;
Step 2: Connect to the external knowledge base to retrieve the proper definition of idioms
<b>For all</b> $i \in \mathcal{D}$ <b>do</b>
<b>For all</b> $j \in \mathcal{F}$ <b>do</b>
$\varphi \leftarrow \text{Mixup}(\text{Embed}[j, i])$ ;
$x_{temp} \leftarrow x_{temp} \cup \text{WEB.CRAWLER}(\alpha, \varphi)$ ;
$\text{dom}(x_{temp}) := \{ i \in \mathcal{D} \mid x_{temp}^i \neq x_{temp} \}$
$\mathcal{F}_{temp} \leftarrow \mathcal{F}_{temp} \cup \text{DOM}(x_{temp})$
<b>End for</b>
<b>End for</b>
<b>Output</b> := $\mathcal{F}_{temp}$
<b>End procedure</b>

#### **4.7 Data Pre-Processing**

Data pre-processing is an essential step in sentiment analysis as it helps to ensure that the input data is in a suitable format for analysis. In this context, the goal of data pre-processing is to clean up the data by removing irrelevant information, standardizing the format of the text, and normalizing the data. This step is necessary because social media data often contains noise, including emojis, URLs, hashtags, stop words, digits, dates, and other characters that do not contribute to sentiment analysis.

The first step in data pre-processing is to remove any unnecessary URLs from the raw tweets. This step can help to speed up the analysis process by reducing the size of the dataset. After that, the text is normalized using various techniques such as stop word removal, case folding, mapping exceptional values to their respective types, special character removal, normalization of acronyms and abbreviations, and spell checking.

Stop word removal involves removing words that are commonly used in a language but do not contribute to the overall meaning of the text. This step can help to reduce the noise in the dataset and improve the accuracy of sentiment analysis. Case folding involves converting all the words in the text to lowercase. This step can help to standardize the format of the text and reduce the complexity of the analysis process.

Mapping exceptional values to their respective types involves identifying and replacing exceptional values such as dates, times, and other numerical values with their respective types. This step can help to reduce the noise in the dataset and ensure that the data is standardized for analysis. Special character removal involves removing hashtags, digits, punctuation marks, and other non-

alphabetic characters from the text. This step can help to simplify the analysis process and improve the accuracy of sentiment analysis.

Normalization of acronyms and abbreviations involves identifying and replacing commonly used acronyms and abbreviations with their full forms. This step can help to ensure that the text is standardized for analysis and reduce the noise in the dataset. Spell checking involves identifying and correcting any spelling errors in the text. This step can help to improve the accuracy of sentiment analysis by ensuring that the text is correctly interpreted by the analysis algorithms.

In summary, data pre-processing is a crucial step in sentiment analysis, and it involves cleaning up the data by removing irrelevant information, standardizing the format of the text, and normalizing the data. The various techniques used in data pre-processing can help to reduce the noise in the dataset and improve the accuracy of sentiment analysis.

Researchers often begin with cleaning the dataset in an attempt to speed up the analysis process. They believe that noisy data is useless and won't improve the system's accuracy. In this thesis, we test this point of view to determine whether it holds water under all circumstances, thereby completing the fourth and final goal of the research. To create a clean baseline sample, we first eliminated any unnecessary URLs from the raw tweets and then normalized the text using the following techniques:

- Stop word removal: removing unnecessary encoding of words absent from any pre-trained word embedding.
- Case folding is the process of converting words or phrases to

lowercase.

- Mapping exceptional values to their respective types (for instance, ‘9 am’→ ‘TIME’),
- Special character removal: removing hashtags, digits, punctuation marks, and other non-alphabetic characters.
- Normalization of acronyms and abbreviations (e.g., ‘UK’→ United Kingdom, ‘idk’→I don’t know).
- Spell checking.

#### **4.8 Evaluation Measures**

The focus in text classification is on evaluating a classifier's effectiveness rather than its computational efficiency. This means that correctly predicting the class of unseen data is the primary concern, instead of the classifier's computational complexity [199]. When sentiment classification is applied to a test dataset, it can yield four possible results: True Positive (TP) for correctly classified positive instances, True Negative (TN) for correctly classified negative instances, False Positive (FP) for incorrectly classified positive instances, and False Negative (FN) for incorrectly classified negative instances. A confusion matrix displaying these four values is presented in Figure 4.5.

		Predicted	
		Negative (N) -	Positive (P) +
Actual	Negative -	True Negative (TN)	False Positive (FP) Type I Error
	Positive +	False Negative (FN) Type II Error	True Positive (TP)

**Figure 4.5: Confusion Matrix for Binary Classification**

To assess how well a classifier works on unseen test data, numerous assessment procedures are utilized. Precision, recall, F-measure, and accuracy are text classification most often used metrics. Frequently, the objective is to maximize all metrics ranging from 0 to 1. As a result, larger numbers indicate greater classification performance. Precision, recall, F-measure, and accuracy are metrics used to evaluate the performance of machine learning models in classification tasks. Here are brief explanations of each metric:

- **Precision:** Precision is the ratio of correctly predicted positive instances to the total number of predicted positive instances. It measures the accuracy of positive predictions, or how many of the predicted positive instances are actually positive. A high precision means that there are few false positives, or instances that are predicted as positive but are actually negative.
- **Recall:** Recall is the ratio of correctly predicted positive instances to the total number of actual positive instances. It measures the completeness of positive predictions, or how many of the actual positive instances are correctly predicted. A high recall means that there are few false



negatives, or instances that are predicted as negative but are actually positive.

- **F-measure:** F-measure is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall. The F-measure is calculated as  $2 * ((\text{precision} * \text{recall}) / (\text{precision} + \text{recall}))$ . It ranges from 0 to 1, with 1 being the best possible value.
- **Accuracy:** Accuracy is the ratio of correctly predicted instances to the total number of instances in the dataset. It measures the overall correctness of the model's predictions. A high accuracy means that the model is making correct predictions for most instances in the dataset. However, accuracy can be misleading if the dataset is imbalanced or if there are other issues such as mislabeled instances.

As well as their widespread usage in text categorization, precision and recall are often used together to gauge the effectiveness of information retrieval systems. The proportions of relevant to irrelevant materials are used to determine “classic” precision and recall. They also factor in the importance of materials that could not be found. Recall measures how many documents were successfully retrieved whereas precision indicates how many were recovered with a high degree of accuracy. Both values might be calculated using equation 4.2.

$$Precision (P) = \frac{TP}{TP+FP}, \quad Recall(R) = \frac{TP}{TP+FN} \quad (4.2)$$

In general, it's very rare to compute the precision and recall independently. The F-measure is a common method of combining the two into a single, more comprehensive metric. The harmonic mean of precision and recall as shown in equation 4.3 can be used to get the F-measure.

$$F - measure = 2 * \frac{P * R}{P + R} \quad (4.3)$$

Others use accuracy as a criterion to assess a classifier's performance. Accuracy (equation (4.4)) is the number of successfully classified occurrences from the total classification predictions. A well-known problem with using accuracy as a measure of a classifier's efficacy is the “accuracy paradox conundrum”. This case happens when the classifiers may achieve high accuracy even if it consistently predicts the same class. This case defies the point of constructing a classifier.

$$Accuracy = \frac{correct\ predictions}{total\ predictions} = \frac{TP + TN}{TP + FN + FP + TN} \quad (4.4)$$

We calculated the error ratio as the complement of the accuracy (Error Ratio = 1 - accuracy) to have a better grasp of the error percentage when annotating and classifying the idiomatic expressions. The  $\delta$  in equation 4.5 denotes the error ratio, and the  $v_A$  and  $v_E$  are the actual observed and the predicted values respectively

$$\delta = \left| \frac{v_A - v_E}{v_E} \right| \cdot 100\% \quad (4.5)$$

#### 4.9 Summary

This chapter presents an overview of the research paradigms, techniques, procedures, and methods that have been used. It gives the reasons for choosing a suitable research approach for the current study that is being done. In addition,

this chapter provides an overview of the various research approaches as well as an argument for why transformers was chosen for this particular piece of work.

In addition, a comprehensive research strategy for the current study has been laid out in this chapter for readers to peruse. The current investigation was carried out in two stages, the first of which was the creation of the model, followed by its validation.

For the purpose of providing a solid foundation for this study, the systematically organized methodologies together with the thorough guidelines and suggestions have been explored. This chapter has a comprehensive explanation of the processes that are involved in each step, as well as design considerations, analytic methodologies, and ways for interpreting the findings. In conclusion, the concerns connected to the idiomatic lexicon creation and annotation have been explored, along with the enrichment strategy for tweet sentiment classification.

## CHAPTER 5

### EXPERIMENTAL RESULTS

To answer the research questions, we conducted several experiments to demonstrate the advantages of using the idiomatic expressions to assess the sentiment polarity of a tweet. Thanks to the expansion method, the overhead required for retraining and fine-tuning the transformer is no longer necessary. To build ground truth data from the tweets dataset, we manually labeled the tweets collection by a group of volunteer annotators. We set criteria to accept tweets that all annotators agreed on the same label, and we collected 3,000 tweets successfully. The sentiment annotation distribution of the tweets is shown in Table 5.1.

**Table 5.1: Sentiments' Polarity Distribution of Ground Truth Tweets Dataset**

<b>Polarity</b>	<b># of tweets</b>
Positive	711
Negative	1,842
Neutral	447

#### 5.1 Experiment I: Automated Idiom Annotation Process

This process refers to the use of NLP techniques and computational methods to automatically identify and annotate idiomatic expressions within a text corpus. This process typically involves identifying multiword expressions that have a non-literal meaning or cannot be interpreted by looking at the individual words alone. The process may involve using external resources such as online dictionaries, thesauruses, or idiom lexicons to support the

identification and annotation of idiomatic expressions. The resulting annotated lexicon can be used for various natural language processing tasks such as sentiment analysis, machine translation, and information retrieval.

The goal of this experiment is to answer the first research question in an attempt to achieve the first three objectives.. RO1: Develop an automated method for building and annotating a sentiment lexicon of idiomatic expressions to eliminate the need for manual annotation. RO2: Investigate and compare different methods for incorporating idiomatic expressions as features in sentiment classification of tweets to determine the most effective approach. RO3: Evaluate the impact of leveraging external knowledge bases on the performance of sentiment analysis of tweets containing idiomatic expressions. We aim to validate the automatic annotation using the deep learning transformer and compare the results to that in the manually annotated gold standard lexicon. This experiment consists of two parts. The first part is conducted using roBERTa as a classifier and the raw idioms (without expansion) as an input. roBERTa is a transformer-based neural language model, which is an improvement over the original BERT model. The roBERTa model was introduced by Facebook AI Research in 2019 and is pre-trained on a large corpus of text to learn general language representations [44]. The roBERTa model is similar to BERT in many ways but includes some notable improvements. Firstly, roBERTa uses a larger pre-training corpus than BERT, which includes more diverse text sources and removes some of the biases present in the original BERT training data. Secondly, roBERTa uses dynamic masking during pre-training, which means that the model is presented with

different masked tokens during each epoch of training. This helps the model to learn more robust representations of language [46].

Another key difference between roBERTa and BERT is in the training procedure. RoBERTa trains on longer sequences of text than BERT, which helps the model to capture longer-term dependencies and contextual information. Additionally, roBERTa uses a larger batch size during training, which enables the model to learn more efficiently and accurately [46].

Overall, roBERTa is an improvement over BERT in terms of its pre-training procedure, training data, and architecture. These improvements result in a model that achieves state-of-the-art performance on a wide range of natural language processing tasks, including question answering, text classification, and natural language inference [44].

The second part is similar to the first one except that idioms are augmented by the different augmentation methods as well as by the idiom expansion method to substitute idioms by their definition/meaning phrases that are retrieved from the external Thesaurus/Dictionary. The second part will answer the fourth research question and achieves the last research objective - RO5: Investigate the influence of data augmentation and normalization pre-processing procedures on the accuracy of the sentiment classifier..

### **5.1.1 Idiom Augmentation and Expansion**

#### **Thesaurus**

Although this method can improve the text data by adding synonyms, it can corrupt the actual meaning if applied to idioms. In the following example, we can see that the method successfully augments “awesome” with “stunning.”

It is an awesome hotel. →  → It is a stunning hotel.

However, in the following example, the definition of the idiom “*bite the bullet*” in the Cambridge dictionary [203] is found as: “force someone to do something unpleasant or difficult, or to be brave in a difficult situation.” If we apply this simple augmentation method and it happens that “bullet” was the randomly chosen word and replaced with “ball” the meaning is changed to “to plagiarize.” In this case, both synonyms happened to have “negative polarity”. However, in other cases, this might wrongly change the sentence polarity.

### **Semantic Embedding**

For the generic word embedding replacement, we notice that the method can successfully replace “amazing” with “incredible” as shown in Figure 5.1. However, if we get back to our idiom example, we note that the word “bullet” is replaced by a “projectile” word which will then change the actual holistic meaning of the idiom.

### **Masked Language Model**

As mentioned in previous chapters, the masked language model was one of the objective tasks that BERT was pretrained for. Although this method successfully predicts the “bullet” word (above part of Figure 5.2), it fails in many other cases. For example, the idiom “*someone had a blast*” in “*I had a blast last night at Trump rally*” represents a positive sentiment. However, this method will the word “fight” as the masked token which will change the whole sentiment to negative.

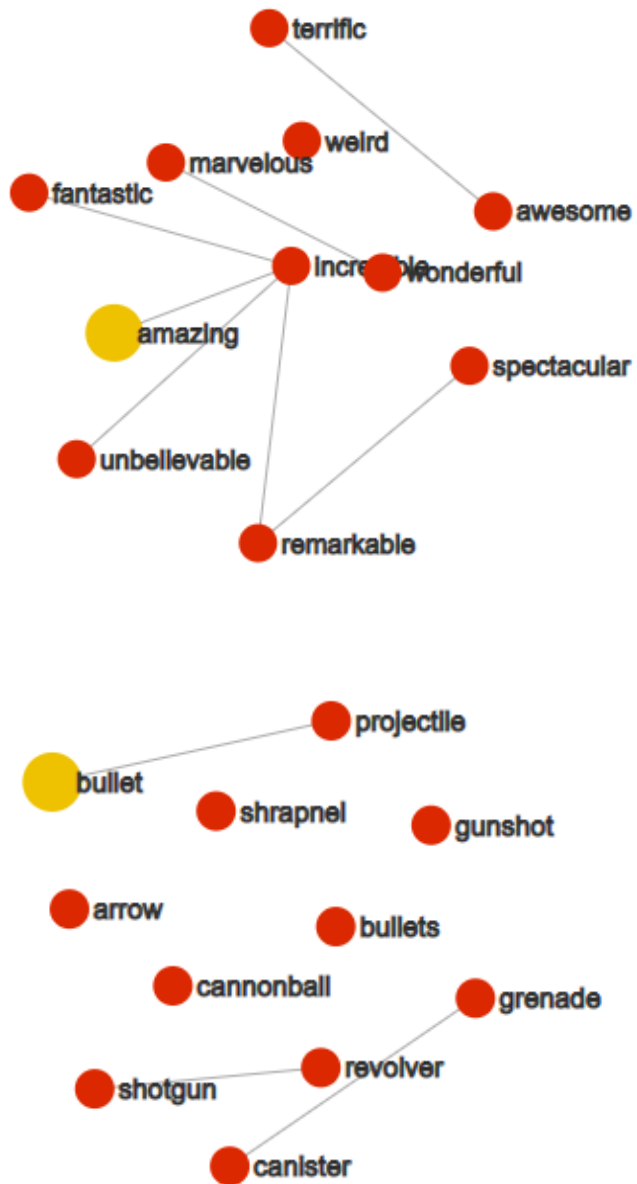
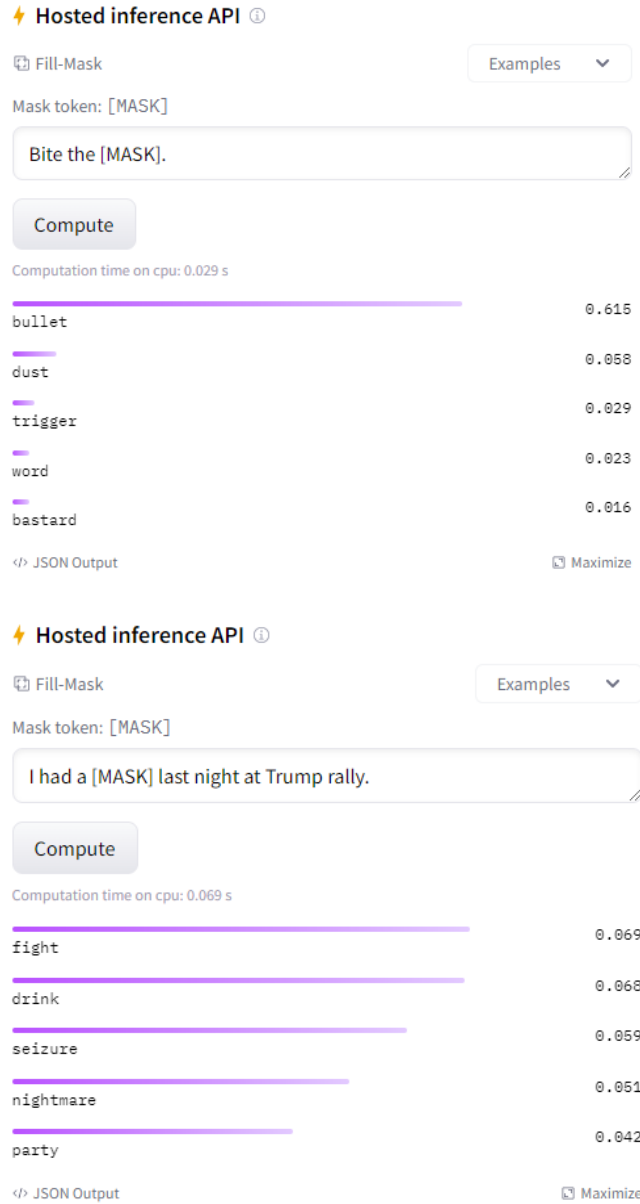


Figure 5.1: The Semantic Relatedness of Terms in the Word2Vec Embedding Space





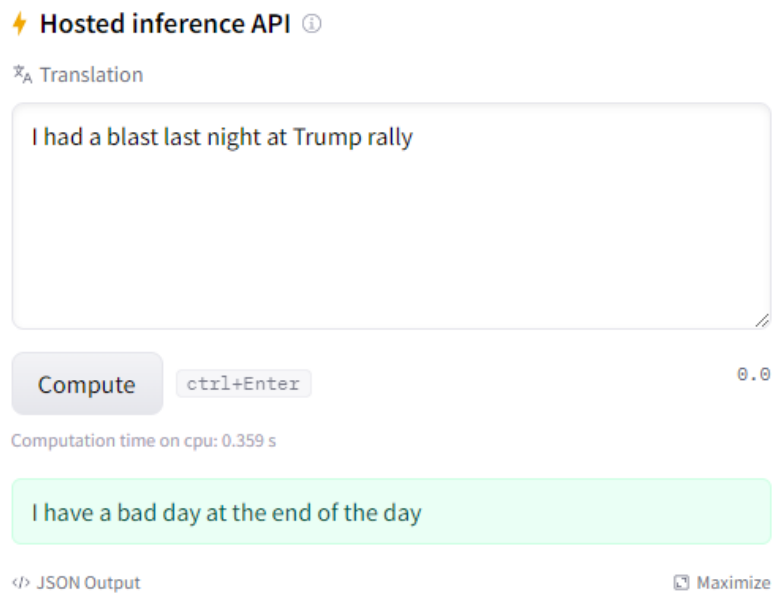
**Figure 5.2: The Output of the MLM Task using BERT**

## Machine Translation

This method enhances sentiment classification and improves the underfitting problem. However, it does not help in the case of idioms. In the following example, translating the tweet “I had a blast last night at Trump rally” was translated to Indian using an Opus-MT model that was fine-tuned on

ai4bhart Hindi-English parallel corpora (SAMANTAR). The Opus-MT model is a machine translation model developed by the Open Parallel Corpus (OPUS) project [204]. It is a neural machine translation model that uses an encoder-decoder architecture with attention mechanisms to translate text from one language to another.

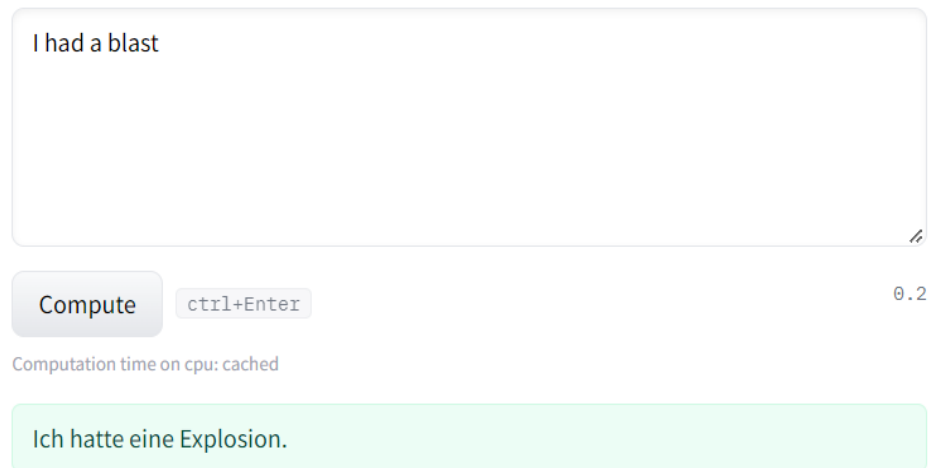
The original English tweet sentiment has a positive sentiment. However, the back-translated result shows an opposite meaning and a strong negative sentiment as shown in Figure 5.3.



**Figure 5.3: Transformer-Based Back-Translations (English-Hindi-English)**

We further test the back-translation between English and German using the opus-mt-de-en model API on the huggingface service website [205]. The example in Figure 5.4 shows the translation result of the “I had a blast” idiom which was translated into German as “Ich hatter eine Explosion.” As we note, the original idiom has a positive sentiment label in the lexicon, however, after the back-translation, the idiom was translated into “I had an explosion,” which

has a negative sentiment.



I had a blast

Compute ctrl+Enter 0.2

Computation time on cpu: cached

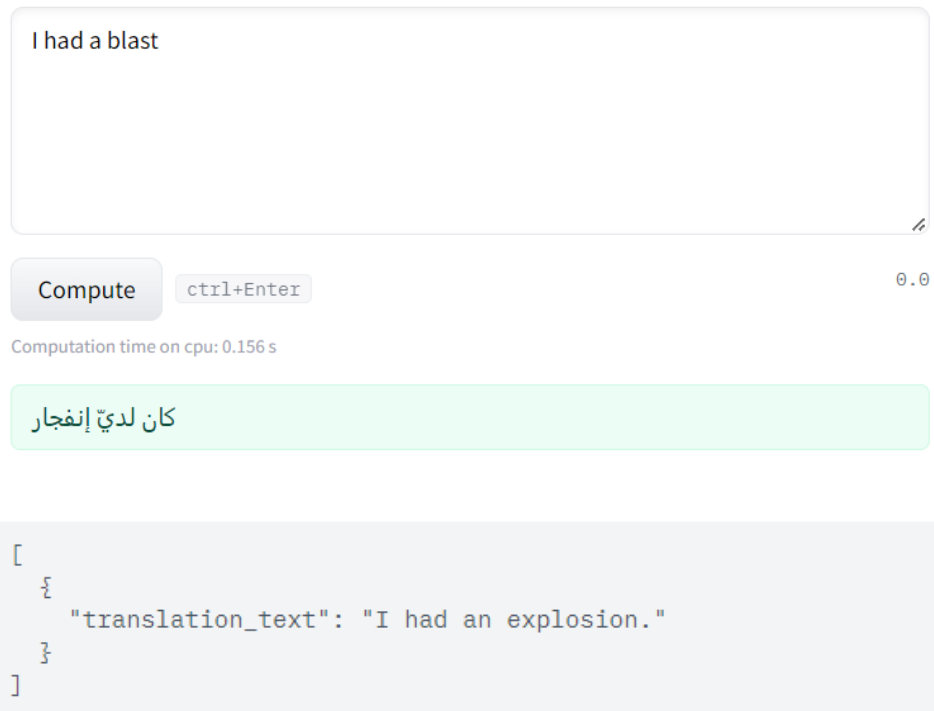
Ich hatte eine Explosion.

```
[
  {
    "translation_text": "Ich hatte eine Explosion."
  }
]
```

```
[
  {
    "translation_text": "I had an explosion."
  }
]
```

**Figure 5.4: Transformer-Based Back-Translations (English-German-English)**

After trying the third language translation, we confirm that the back-translation method for idiomatic expressions is not helpful. In this test, we use English-Arabic-English back-translation and the result is shown in Figure 5.5. The back-translation from English to Arabic gives the “I had an explosion” sentence with negative sentiment.



**Figure 5.5: Transformer-Based Back-Translations (English-Arabic-English)**

### Idiom Expansion

Using API provided by online resources such as Oxford Dictionaries and The Free Dictionary websites, we retrieve the definition of idioms by selecting the first definition (some websites provide more than one definition).

By referring to idioms in the lexicon, we formulate the search query by iterating over each idiom. Since most idioms have more than one definition in the same thesaurus/dictionary, we retrieve the first definition only. Table 5.2 shows the retrieved definitions of the provided sample idioms.

**Table 5.2: Sample of Idiom Expansion from Two Different Resources**

<b>Idioms</b>	<b>Oxford dictionaries</b>	<b>The fee dictionary</b>
<i>kick the bucket</i>	die	to die
<i>in dutch</i>	in trouble	in trouble
<i>keyed up</i>	nervous, tense, or excited, especially before an important event	excited, nervous, or anxious.
<i>lay an egg</i>	be completely unsuccessful; fail badly	fail
<i>life of riley</i>	N/A	a life of great ease, comfort, or luxury,
<i>on the rocks</i>	experiencing difficulties and likely to fail	served undiluted and with ice cubes.
<i>Achilles' heel</i>	a weakness or vulnerable point	a fatal weakness, a vulnerable area

### 5.1.1.1 Lexicon Annotation using Expansion Method

SLiDE provides a sentiment lexicon of idiomatic expressions consisting of 5000 annotated idioms. Our extension led to an increase in the number of idioms in the lexicon which become 8,930 idioms. Table 5.3 shows the major columns of a sample from the manually annotated lexicon. An idiom is assigned a sentiment class label based on the majority votes it receives.

**Table 5.3: Sample of eSLiDE Lexicon Automatic Annotation of Idioms Using Twitter-Roberta-Base-Sentiment Classifier without Expansion**

<b>Idiom</b>	<b>Pos</b>	<b>Neg</b>	<b>Neu</b>	<b>Maj. Label</b>
kindred spirit	0.5	0	0.5	positive
killer instinct	0.3	0.2	0.5	neutral
kettle of fish	0	0.5	0.5	negative
jump through hoops	0.3	0.2	0.5	neutral
jump on the bandwagon	0.5	0	0.5	positive
jack of all trades	0.4	0.1	0.5	neutral
inside track	0.5	0	0.5	positive
in the dark	0	0.5	0.5	negative
in the bag	0.5	0	0.5	positive
in a pinch	0	0.5	0.5	negative
ice cool	0.5	0	0.5	positive
house of cards	0	0.5	0.5	negative

We follow the same annotation priority that was used in SLiDE; when two labels have the same percentage of votes, the annotation priority is set based on the following rule: The label with a higher percentage number has higher priority as Negative > Positive > Neutral.

In Table 5.4, we present the automated annotation results using the roBERTa transformer. We access the Twitter-roberta-base-sentiment model from the hugging face library [206]. It's important to note that this model was utilized since it was pretrained to handle the sentiment analysis task. We neither retrain nor fine-tune this model but rather access it directly using the API provided by the hugging face website. It's worth noting that unlike the rigged manual annotation, where we count the number of votes, the model produces a probability-scoring percent of how much an idiom belongs to a sentiment class. We can compute the annotation error ratio before and after the idiom expansion by simply counting the mismatched labels.

In table 5.5, the error ratio was computed for each sentiment class. We can note that the expansion method has sharply dropped the error ratio, especially for the positive sentiment class. The preliminary result indicates the usefulness of using the expansion method. However, we do further analysis to see if we can improve the annotation accuracy. Therefore, rather than retrieving the first definition found in the external dictionary/thesaurus, we retrieve the definition randomly. We note from Table 5.6 that this method has reduced the error ratio and moved the F1-Score from 89.8% to 95.4%.

**Table 5.4: Sample of Twitter-Roberta-Base-Sentiment Sentiment Classification With/Out Idioms Expansion**

Idiom’s label probability score with/out idioms expansion						Comparison with idioms labels in SliDE		
Pos%		Neg%		Neu%		Matching with Lexicon?		Label in SliDE
0.163	0.083	0.081	0.04	0.754	0.878	No	No	Pos
0.104	0.453	0.209	0.028	0.686	0.519	Yes	Yes	Neu
0.095	0.016	0.226	0.69	0.678	0.294	No	Yes	Neg
0.073	0.035	0.184	0.125	0.743	0.841	Yes	Yes	Neu
0.207	0.712	0.135	0.004	0.659	0.284	No	Yes	Pos
0.077	0.573	0.247	0.009	0.676	0.418	Yes	No	Neu
0.129	0.51	0.131	0.017	0.74	0.473	No	Yes	Pos
0.107	0.015	0.172	0.796	0.721	0.189	No	Yes	Neg
0.228	0.634	0.111	0.044	0.661	0.323	No	Yes	Pos
0.158	0.016	0.149	0.701	0.694	0.283	No	Yes	Neg
0.763	0.154	0.017	0.692	0.22	0.154	Yes	No	Pos
0.101	0.006	0.117	0.201	0.782	0.793	No	Yes	Neg

**Table 5.5: Comparison of Error Ratio While Annotating Idioms Without-Out Idiom Expansion**

$\delta$ of annotating sentiment class	Without expansion	With expansion
Positive label	38%	11%
Negative label	24%	13%
Neutral label	16%	6%

**Table 5.6: Precision, Recall, and F1-Score Results Using roBERTa**

Automatic idiom annotation performance using roBERTa			
Metric	Raw Idioms	Using Expansion with the first definition	Using Expansion with multiple definitions
<b>Precision</b>	0.733	0.883	0.942
<b>Recall</b>	0.752	0.913	0.966
<b>F1-score</b>	74.2%	89.8%	95.4%

### 5.1.1.2 The Impact of Different Augmentation and Expansion Methods

In the last part of this experiment, we compare the F1-Score for every

augmentation method and present the results of the expansion method in Table 5.7. It’s incredibly noted that the highest F1-Score was achieved by the proposed idioms-expansion method with a value above 95%. However, we notice that some of the augmentation methods (Masked Language Model, Machine Translation, and substitution) have degraded the classification performance and received a lower score than even the usage of the raw idioms without any augmentation.

**Table 5.7: Annotation of the Sentiment Lexicon of Idiomatic Expressions Using the Twitter-Roberta-Base-Sentiment Transformer and Different Augmentation Methods**

<b>Augmentation Method</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>
Thesaurus	0.841	0.855	0.847942
Semantic Embeddings	0.799	0.814	0.806431
Masked Language Model	0.784	0.687	<u>0.732302</u>
Machine Translation	0.705	0.714	<u>0.709471</u>
Swapping	0.795	0.881	0.835794
Deletion	0.814	0.801	0.807448
Insertion	0.742	0.784	0.762422
Substitution	0.687	0.626	<u>0.655083</u>
SentMixup	0.822	0.841	0.831391
Idiom expansion	0.942	0.966	<b>0.953853</b>
<b>Raw idiomatic expressions</b>	0.733	0.752	0.742378

## 5.2 Experiment II: Sentiment Classification of Tweets

### 5.2.1 Sentiment Classification of Tweets Dataset by Idioms Labels

We conduct this experiment to answer the second question to achieve the second research objective RO2: Propose an idiomatic expansion method to enrich the tweet’s context using external knowledge bases. This experiment is employing the tweet enrichment method by considering the positional context of the idiom. For tweet sentiment classification, the experiment is composed of four parts. The first part is to directly assign the sentiment label of an idiom to



the tweet itself. There is no learning used in this part but rather a simple assignment. The error rate of classifying tweets sentiments based on the direct assignment of idioms labels to tweets is shown in Table 5.8. A ratio representing the error rate is obtained by dividing the total number of polarity labels by the total amount of errors, as in equation 5.1.

$$\text{Error rate} = \text{total polarity} / \text{total errors} \quad (5.1)$$

The rate is expressed as 1:19 means that for every 100 errors in classifying tweets, there are 119 tweets identified correctly using the direct assignment of the idioms' labels.

**Table 5.8: Error Percentage of Direct Sentiment Assignment Based on Idiom Label**

<b>Polarity</b>	<b># of tweets in each sentiment class</b>	<b>#of tweets having the same sentiment as the idioms it contains</b>	<b>ER</b>	<b>Accuracy (1- <math>\delta</math>)</b>
Positive	711	114	1.19	16%
Negative	1,842	790	1.75	43%
Neutral	447	380	6.67	85%

### 5.2.2 Sentiment Classification of Tweets Dataset using roBERTa

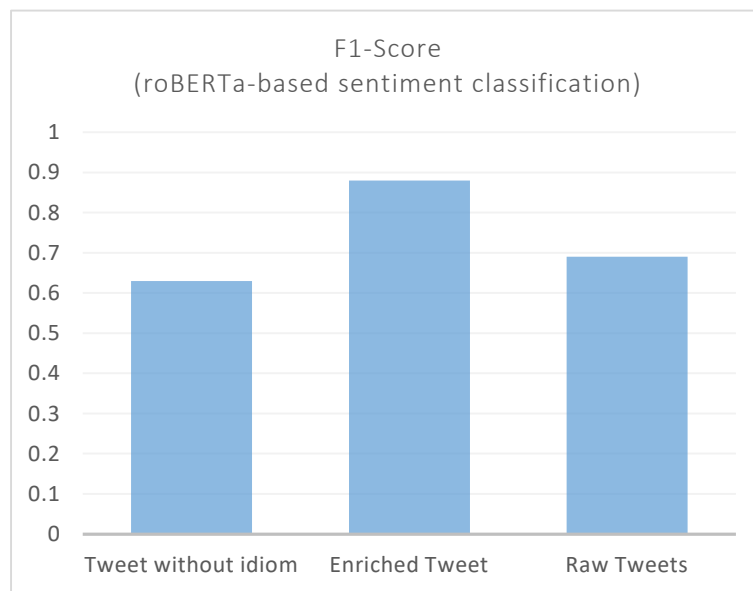
The second part of the experiment aims to measure the accuracy and the error rate associated with feeding the transformer with the raw tweets (including idioms) without any expansion.

In the third part, the aim is to measure the influence of idioms on the overall sentiment classification of the tweet. In this part, we remove the idiomatic expressions from the tweets and evaluate the performance using various classification metrics. In the last part, we replace the idioms with their

definitions at their positional location and evaluate the classifier performance using various metrics. The accuracy result is shown in Table 5.9 and the F1-score result is depicted in Figure 5.6.

**Table 5.9: Accuracy of Tweet Sentiment Classification Using the Twitter-Roberta-Base-Sentiment**

roBERTa Classifier Accuracy			
Test Benchmark dataset 3000 tweets	Omitting idioms from a tweet	Raw vs. Enriched Tweets	
		1,842 Neg	69%
711 Pos	72%	70%	89%
447 Neu	47%	67%	85%



**Figure 5.6: F-1 Score Results of roBERTa-Based Sentiment Tweet Classification**

### 5.2.3 Sentiment Classification Accuracy among Annotators

We operated on the presumption that the sentiment polarity of a tweet

should be the same as that of the idioms it includes. Having said that, this presumption may only be correct if the context of the tweet in question has the potential to influence either the meaning or the aim of an idiom. Unfortunately, it would be impossible to manually annotate the remainder of the unlabeled tweets to use them as a benchmark reference and directly calculate the accuracy based on our assumption. Because of this, we use our approach to automatically annotate the whole of the dataset according to the polarity of the emotion, and then we ask each annotator to manually annotate a subset consisting of 500 random tweets. In this experiment, instead of using a “majority label” as we did in the previous one, which was based on the inter-annotator agreement, we instead swap the pair 500-tweet subsets between every two annotators. This allows us to avoid any potential bias that might be introduced by the inter-annotator agreement. This technique has the potential to provide us with some insight into the general accuracy and consistency of the automated annotation in comparison to the manual annotation. Table 5.10 illustrates how accurate the sentiment classification of tweets was across all of the different subsets of this experiment.

**Table 5.10: Accuracy and Consistency of the Tweets Classification Using the Automatic-Based Annotation Method**

500 tweets datasets	Annotator ID	Sentiment Classification Accuracy
		Difference
Dataset 1	1	89%
	2	88%
		<b>1%</b>
Dataset 2	2	92%
	1	92%
		<b>0%</b>
Dataset 3	3	94%
	4	96%
		<b>2%</b>
Dataset 4	4	95%
	3	95%
		<b>0%</b>
Dataset 5	5	88%
	6	90%
		<b>1%</b>
Dataset 6	6	87%
	5	87%
		<b>0%</b>
Dataset 7	7	79%
	8	85%
		<b>6%</b>
Dataset 8	8	83%
	7	85%
		<b>2%</b>
Dataset 9	9	93%
	10	93%
		<b>0%</b>
Dataset 10	10	94%
	9	93%
		<b>1%</b>
<b>Average accuracy</b>		<b>90%</b>

### 5.3 Experiment III: Classification Comparison of Different Deep Learning Methods

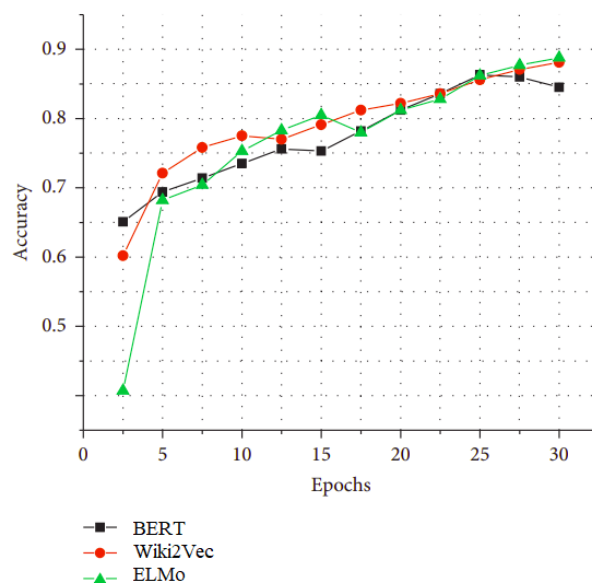
We conduct the last experiment to answer the third research question and to achieve the RO4: Propose and build a classification model combining lexicon and deep learning methods to classify sentiments concealed in tweets' datasets. We used different embedding representations produced from different word embedding methods to test the F-score of LSTM, CNN, and hybrid CNN-LSTM deep learning models. The idea behind embedding assumes that similar words

and their contexts have similar vector representations, as in the sentences “Best Italian restaurant in Kuala Lumpur” and “Top Italian food in Kuala Lumpur”. Most likely both sentences will be used in a similar context and also with similar words that have similar vector representations, such as “pasta”, “favorite”, or “cuisine”. These vectors were used as the input for the deep learning models.

We utilize the deep learning models and compare their performance with the roBERTa transformer using different embedding of the enriched tweets at both word and sentence vector representations. After tuning the deep learning models, we found the best hyperparameters for each model, i.e., the ones that achieved the highest F-score. Soft voting is then utilized to average the projected probability of class membership among the selected models, and the hybrid class prediction was picked from the class with the highest average. In soft voting, the predicted probabilities of each model are averaged or weighted to obtain a final predicted probability for each class. This is in contrast to hard voting, where the final prediction is made based on the most frequently predicted class by each model. The soft voting method is used to combine the predicted probabilities of the deep learning models, which have been trained on enriched tweets using different word and sentence vector representations, and the roBERTa transformer model. The resulting hybrid class prediction is then made by selecting the class with the highest average predicted probability across the selected models. The use of soft voting can improve the accuracy and reliability of classification tasks by taking into account the individual strengths and weaknesses of each model and combining them in a more nuanced way than hard voting.

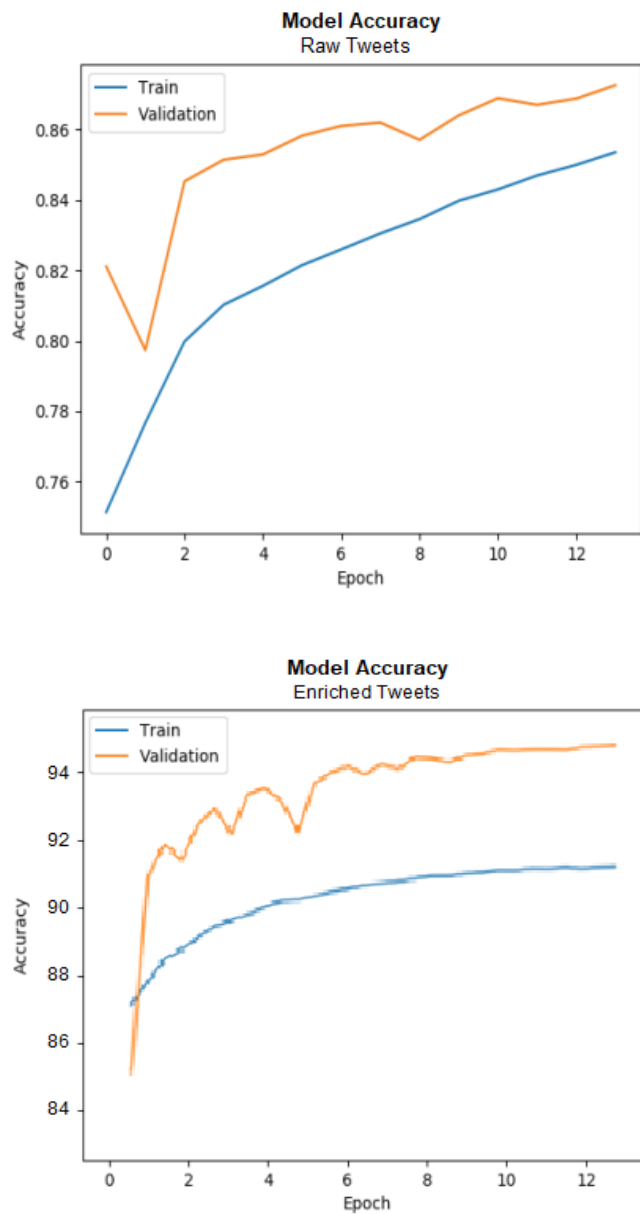
To train the deep learning models, we use the Sentiment140 dataset [207]. As shown in Table 5.11, the obtained results showed that the classification of tweet sentiment achieved a better F1 score when using the enriched tweets over the raw baseline tweets. The last column describes the accuracy differences that were gained, noted as  $\text{Gain}_{\text{percentage}} = \text{Fscore}_{\text{enriched}} - \text{Fscore}_{\text{raw}}$ .

Interestingly, almost all word-level embeddings performed better than sentence-level embeddings on the given data set. However, regardless of which embedding technique was used, the accuracy of the proposed enrichment data representations outperformed the raw baseline data set. Based on Table 5.11, BERT, Wiki2Vec, and ELMo were the best-performing embedding techniques, with an accuracy percentage close to 90%. Figure 5.7 shows a comparative analysis of the accuracy/epoch curves for the tweet enrichment method with various embedding representations.



**Figure 5.7: Accuracy of BERT, Wiki2Vec, and ELMo Embedding Methods as the Number of Epochs Increases (enriched tweets)**

From Figure 5.8 we may infer that the tweet enrichment procedure was critical in gaining access to factors that could not be determined straight from the raw tweet data. In Figure 5.9 we present BERT embedding and compared the results of using baseline vs. enriched data representations. We can conclude that the enrichment model may be used to improve sentiment analysis on a larger scale.



**Figure 5.8: Accuracy of the Hybrid CNN-LSTM Model Using the ELMo Embedding**

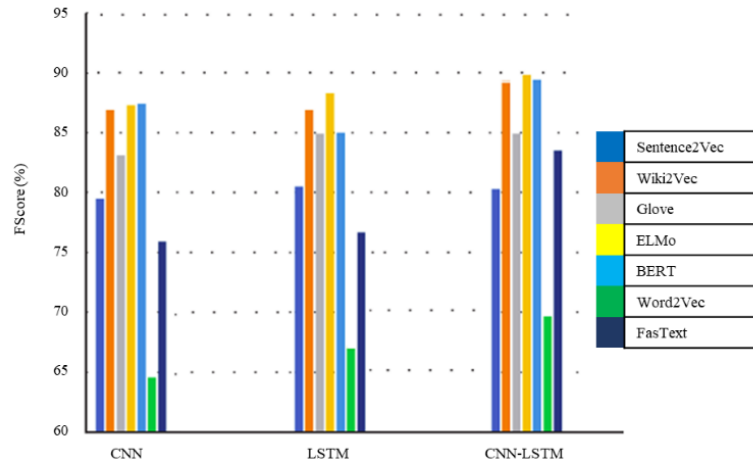
After evaluating the performance of the proposed tweet enrichment method, we can be concluded that the tweet enrichment method performs well with all embedding techniques. Therefore, the proposed enrichment method is effective in improving sentiment classification results. We may also conclude that the hybrid CNN/LSTM model improves the accuracy of sentiment analysis compared with the individual learning models.

Although the hybrid model achieved a comparable F1-score with the roBERTa transformer (both around 90 points), the hybrid model still requires training and tuning which requires time and computing power, unlike the roBERTa transformer that we utilize without fine-tuning, thanks to the enrichment method.

**Table 5.11: Precision, Recall, and F-score Comparison of Raw and Enriched Tweets**

Embedding method	Raw vs. Enriched Tweets						Gain percentage
	Precision		Recall		F-Score		
<b>Word2Vec</b>	0.47	0.53	0.45	0.66	0.46	0.59	13%
<b>Sentence2Vec</b>	0.69	0.75	0.70	0.82	0.69	0.78	9%
<b>Glove</b>	0.71	0.80	0.76	0.82	0.73	0.81	8%
<b>FastText</b>	0.63	0.73	0.71	0.87	0.67	0.80	13%
<b>BERT</b>	0.74	0.84	0.81	0.96	0.76	0.89	13%
<b>ELMo</b>	0.74	0.87	0.79	0.93	0.76	0.90	14%
<b>Wiki2Vec</b>	0.71	0.86	0.8	0.92	0.75	0.89	14%





**Figure 5.9: Comparison of the F-Score Results of the CNN-LSTM Hybrid Model, CNN, and LSTM Concerning Different Embedding Techniques**

As was previously indicated, certain idioms are ambiguous by nature and may have a muddled “sentiment” of their own. These idioms change depending on the context they appear in. For instance, the expression “tough as nails” may mean either “manage any difficulty” or “be cruel and unfeeling”. Another example is the expression “well-padded,” which may also mean either “being rich” or “being fat,” and depending on the context in which it appears, can either be taken as praise or as an insult. Even human annotators struggle to identify which polarities to assign for idioms supplied to them in isolation.

#### 5.4 Experiment IV: Handling the Confusing Idioms

We conduct another experiment to find out whether a bipolar idiom can be appropriately categorized based on a definition chosen at random or by combining all definitions if there are several meanings available. For this experiment, we have identified 150 tweets with different idioms that might have bipolar definitions or meanings. Table 5.12 demonstrates that introducing random definitions results in inconsistent behavior, with some idioms being incorrectly classified while others benefit from being properly classified. This

is shown by the fact that certain idioms are incorrectly classified. On the other hand, the idiom expansion utilizing the multi-definitions fusion approach was only successful in one out of the four attempts to improve the F-score metric.

The selection of the single definition is done randomly for each run (not the first definition found in the thesaurus or the dictionary). As can be seen in Table 5.12, the accuracy of the multi-definition fusion and the no-expansion settings were maintained across all of the many iterations of the experiment by not being altered in any way.

**Table 5.12: F1-Score Comparison of the Expansion Method Using Single and Multi-Definition Methods**

<b>Sentiment Classification (150 annotated Tweets) using roBERTa</b>		
<i>F1-Score</i>		
<i>Baseline</i>	<i>Expansion using the random definition</i>	<i>Expansion using a fusion of all definitions</i>
76.98%	89.42%	88.35%
76.98%	81.68%	88.35%
76.98%	93.82%	88.35%
76.98%	91.47%	88.35%

### **5.5 Experiment V: Performance Comparison of Lexicon Annotation using Roberta and Off-Self Sentiment Classification Tools**

In the last experiment, we compare the roBERTa classifier to those off-shelf sentiment classification tools. Mainly the coreNLP and SentiStrength tools. Table 5.13 illustrates sample idioms from Slide and their respective annotations before and after expansion. We can see from Table 5.14 that the roBERTa transformer outperforms the other two classifiers, however, we can see that the idiom expansion has notably improved the sentiment annotation for all three methods.

**Table 5.13: Sample of the Sentiment Lexicon Annotation before Expansion Using Different off-shelf Sentiment Classifiers**

Idiom	Manual label	Idiom Classification match (before/after) expansion					
		roBERTa		CoreNLP		SentiStrength	
catch-22	Neg	F	<b>T</b>	F	F	F	<b>T</b>
prince of darkness	Neg	F	<b>T</b>	F	F	F	F
comedy of errors	Neg	<b>T</b>	<b>T</b>	F	F	F	F
cook the books	Neg	F	<b>T</b>	F	F	F	F
cop-out	Neg	F	F	F	F	F	F
cough up	Neg	F	F	F	F	F	F
fucked up	Neg	F	<b>T</b>	<b>T</b>	<b>T</b>	F	F
god forbid	Neg	<b>T</b>	<b>T</b>	F	<b>T</b>	F	F
command performance	Neg	F	F	<b>T</b>	F	<b>T</b>	F
number ten	Neg	<b>T</b>	F	F	F	<b>T</b>	F
christmas present	Pos	<b>T</b>	<b>T</b>	F	<b>T</b>	F	F
cotton to	Pos	F	<b>T</b>	F	F	F	F

**Table 5.14: Idiom Annotation Comparison: roBERTa vs. CoreNLP, and SentiStrength Tools**

	Metric	Method		
		roBERTa	CoreNLP	SentiStrength
Before – after idiom Expansion (Slide idioms)	Accuracy (ACC)	0.72	0.18	0.11
		<b>0.91</b>	<b>0.57</b>	<b>0.36</b>
	Error rate (1-Acc)	0.18	0.82	0.89
		<b>0.09</b>	<b>0.43</b>	<b>0.64</b>
	F1-score	0.71	0.12	0.08
		<b>0.88</b>	<b>0.46</b>	<b>0.29</b>

## CHAPTER 6

### CONCLUSION & FUTURE WORK

#### 6.1 Overview

In conclusion, this research work has contributed towards automating the creation and annotation of a sentiment lexicon of idiomatic expressions. The key findings of this work have been summarized, including the role of idiom expansion and the novelty of this research. Additionally, the benefits of tweet enrichment have been highlighted in improving the sentiment analysis of Twitter's big data, without requiring the retraining and fine-tuning of the Transformer. Finally, future work has been presented in the last section. Overall, this research work provides a valuable contribution towards improving sentiment analysis in social media, and lays the foundation for further research in this area.

#### 6.2 Summary of Results

This thesis's research was motivated by the observation that idioms, despite their significance, are underused as features in sentiment analysis tasks. We expected that when idiom-based features are provided, they would improve sentiment classification. Idioms often represent an emotional orientation toward an item or an event. To determine how much value idioms provide to sentiment analysis, we used them in conjunction with transformer-based and other deep learning-based approaches and examined classification results. Our research showed that using the expansion method for idioms enhances sentiment classification outcomes substantially. However, even if state-of-the-art learning

techniques may infer the correlation and contextual link between the terms of multi-worded expressions, they cannot deduce their true meaning when idioms are used literally.

We extended the SliDE lexicon and make it available to the public to allow other researchers in the NLP community to do more research into various methods of exploiting idioms as features of sentiment analysis. The extension comprises a large collection of nearly 4,000 idioms that have been carefully annotated with sentiment polarity and have a reliable inter-annotation agreement of ( $\alpha = 0.696$ ). To the best of our knowledge, this dataset is one of the biggest sentiment lexicons of idiomatic expressions of its sort that can be used for sentiment analysis tasks.

We simply retrieved a single definition per idiom and use it to reveal the actual sentiment. While this technique is adequate to highlight the importance of idioms in sentiment analysis, it does not ensure consistent results if the definition is changed in the external resource or if we randomly select another idiom. The outcomes of this thesis might be further improved in different ways. First, we can investigate if the concatenation of all definitions might further improve the classifier performance of might bring more consistent results.

Even though the identification of idioms in tweets is not the focus of the research that is being conducted, we have compiled a library of local grammar that may be used to identify occurrences of idioms in tweets. In addition to this, we have built a corpus of 3,000 English tweets that include a variety of idioms that are utilized in context. This corpus, in addition to the idioms, was manually annotated with the polarity of the respective sentiment. As a result, it is possible

to use this corpus in the systematic evaluation of sentiment analysis algorithms that claim to incorporate idioms as features. The performance of sentiment analysis may also be improved by simply expanding the range of idioms that are represented by the vocabulary that was previously discussed. Instead of increasing these resources by hand, we suggested an expansion technique that would automate this portion of the process. This would boost the generalizability of our system and make it possible for it to be ported to other languages. This method solves the fundamental constraint of the knowledge-engineering overhead needed in hand-crafting the lexicon and the human labor necessary in annotating the idioms' sentiment polarity. Therefore, we anticipated that it is feasible to computationally extract sentiment from dictionary definitions of idioms to automate the acquisition of their sentiment polarity.

To validate our hypothesis and assess the practicality of this strategy, we substituted the humanly constructed lexicon with their automatically produced equivalents and performed the classification experiments. When such idiom-based characteristics were available, sentiment analysis classification performance increased, confirming our prediction. Despite producing good results (F1-Score 95%), the method is still inferior to the manually annotated lexicon. However, the completely automated technique has the benefit of repurposing existing idiom dictionaries, enabling an arbitrary lexicon of idioms to be investigated as part of sentiment analysis.

## **6.3 Research Implication**

### **6.3.1 Theoretical Implications:**

- The research contributes to addressing the automation of the

creation and annotation of a sentiment lexicon of idiomatic expression.

- The study highlights the significance of using idioms as features in sentiment analysis tasks.
- The research shows that using the expansion method for idioms enhances sentiment classification outcomes substantially.
- The study confirms that the correlation and contextual link between the terms of multi-worded expressions cannot deduce their true meaning when idioms are used literally.
- The research suggests that the concatenation of all definitions might further improve the classifier performance or bring more consistent results.
- The method of extracting sentiment from dictionary definitions of idioms can automate the acquisition of their sentiment polarity.

### 6.3.2 Empirical Implications:

- The research provides a large collection of nearly 4,000 idioms that have been carefully annotated with sentiment polarity and can be used for sentiment analysis tasks.
- The study provides a library of local grammar that may be used to identify occurrences of idioms in tweets.
- The research builds a corpus of tweets that include a variety of idioms that are utilized in context and manually annotated with the

polarity of the respective sentiment.

- The study suggests an expansion technique that automates the process of increasing the resources of idioms represented by the vocabulary.
- The research shows that sentiment analysis classification performance increased when idiom-based characteristics were available, confirming the prediction that using idioms as features improves sentiment classification.
- The automated technique has the benefit of repurposing existing idiom dictionaries, enabling an arbitrary lexicon of idioms to be investigated as part of sentiment analysis.

Overall, this research contributes to the NLP community by providing a valuable resource for sentiment analysis tasks and highlighting the importance of using idioms as features in sentiment analysis. The study provides empirical evidence that using idioms can substantially improve sentiment classification outcomes and suggests practical strategies for automating the process of creating and annotating sentiment lexicons of idiomatic expressions.

#### **6.4 Future Work**

In this section, we explore potential avenues for expanding upon the findings of this thesis in further study. The evaluation in the experimental chapter highlights the fact that there is room for improvement in the performance of a fully automated method to employing idioms as features in sentiment analysis.



This thesis proposed an idiom expansion method to enrich tweets' context to improve sentiment classification accuracy. The expansion method utilizes an external knowledge base to extract the non-literal meaning of idioms. The technique avoids the instability caused by the conventional transformers' fine-tuning to solve a downstream task. It's worth studying the suggested expansion for autonomous production of an idiomatic sentiment lexicon rather than crowdsourcing services.

To understand the task of classifying the sentiment of tweets using an idiomatic sentiment lexicon, this thesis suggests a special augmentation technique using the idiom expansion method. It employs a full replacement of an idiom with its factual meaning or definition retrieved from an external knowledge source. The objective was to assess the validity and accuracy of using the expanded form of an idiom for both lexicon annotation and sentiment classification using the BERT transformer model. The findings of this study demonstrate that data expansion is quite beneficial and that it can be employed even without fine-tuning the transformer.

Since the BERT transformer beat other algorithms in the annotation process of the original gold standard, we decided to employ it again for the tweet sentiment classification task to ensure compliance with the original research. Due to the unequal distribution (imbalanced annotation classes where neutral label 40%) of the sentiment lexicon, the performance of the SentiStrength may degrade since the implementation of SentiStrength relies on the sentiment of each word and ignores the context. It is also possible that different deep learning approaches or other advanced transformers may result in more/less effective

classification results.

The sensitivity of classification for positive and negative polarity is increased when idioms are included as features, regardless of whether the idioms were personally constructed or automatically produced. It was shown that machine-created annotations were more likely to have a bias toward a negative polarity than ones that were made manually. It's possible that the way the idiom's polarity is encoded is to blame for this.

The initial crowdsourced idiom polarities enabled a hazy depiction of polarities due to the dispersed nature of the annotations across the three places (positive, negative, and neutral). The term “concrete jungle,” for instance, has several meanings depending on the situation, hence it was originally represented by the polarity vector (0, 30, and 70). However, such a depiction is impossible using the idiom polarities that were automatically extracted. The polarity vector for the definition of the “concrete jungle” (*An overcrowded, unsafe, and/or crime-ridden urban environment or city, characterized by the congestion of large buildings and roads*) is (0.003, 0.06, 0.973) indicating that the phrase has an entirely negative connotation. To fix this, either we provide the context or we may modify the idiom polarity representation to include the idea of ambiguity and/or intensity.

Future research will examine the impact of altering training and testing datasets, the embedding model, and the expansion method on the outcomes. The last experiment raised another unanswered question: how can we ensure that the classification model will function well when the expansion procedure is done at random rather than picking the first definition found in the online dictionary?

We think it will be highly worthwhile to go further to address this query by modifying the expansion process to take into account the general sentiment of the tweet itself before choosing the appropriate meaning/definition of the idiom.

The framework only handles three sentiment classes (positive, negative, and neutral). It's very promising to enhance this framework to support multi-label classes like "Optimistic, Thankful, Empathetic, Pessimistic, Anxious, Sad, Annoyed, Denial, Surprise, and Joking," which are presented in a Kaggle dataset [200]. Another research direction is to investigate the impact of fusing more than one definition into a single feature vector and then applying some of the best-performing augmentation methods that are mentioned in the experimental Chapter.

## REFERENCES

- [1] C. Strapparava and R. Mihalcea, “Learning to identify emotions in text,” in *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, 2008.
- [2] C. Strapparava and A. Valitutti, “Wordnet affect: an affective extension of wordnet,” *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, vol. 4, 2004.
- [3] V. Rentoumi, G. A. Vouros, V. Karkaletsis, and A. Moser, “Investigating metaphorical language in sentiment analysis: A sense-to-sentiment perspective,” *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 9, no. 3, pp. 1–31, 2012.
- [4] S. A. Bobrow and S. M. Bell, “On catching on to idiomatic expressions,” *Mem. Cognit.*, vol. 1, no. 3, pp. 343–346, 1973.
- [5] T. Bulut and I. Çelik-Yazici, “Idiom processing in l2: Through Rose-colored glasses,” *The reading matrix*, vol. 4, no. 2, 2004.
- [6] C. Fernando, *Idioms and Idiomaticity*. London, England: Oxford University Press, 1996.
- [7] C. Cacciari and P. Tabossi, “The comprehension of idioms,” *J. Mem. Lang.*, vol. 27, no. 6, pp. 668–683, 1988.
- [8] M. Mosbach, M. Andriushchenko, and D. Klakow, “on the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines.” *The 9th International Conference on Learning Representations*. Austria, 2021.
- [9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional Transformers for language understanding,” in *Proceedings of the 2019 Conference of the North*, 2019, pp. 4171–4186.
- [10] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi *et al.*, “Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping,” *arXiv preprint arXiv:2002.06305*, 2020.
- [11] C. L. Lee, K. Cho and W. Kang, “Mixout: Effective Regularization to Finetune Large-scale Pretrained Language Models,” In *International Conference on Learning Representations*.

- [12] CFI Team, “Figurative language,” *Corporate Finance Institute*, 31-May-2020. Available: [Accessed: 11-Sep-2021].
- [13] C. Jochim, F. Bonin, R. Bar-Haim, and N. Slonim, “SLIDE - a sentiment lexicon of common idioms,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [14] “Number of worldwide social network users 2027,” *Statista*. [Online]. Available: <https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/>. [Accessed: 11-Dec-2022].
- [15] B. Tahayna, R. K. Ayyasamy, and R. Akbar, “Automatic sentiment annotation of idiomatic expressions for sentiment analysis task,” *IEEE Access*, vol. 10, pp. 122234–122242, 2022.
- [16] M. Furini and M. Montangero, “TSentiment: On gamifying Twitter sentiment analysis,” in *2016 IEEE Symposium on Computers and Communication (ISCC)*, 2016.
- [17] B. Tahayna, R. Ayyasamy, and R. Akbar, “Context-aware sentiment analysis using tweet expansion method,” *J. ICT Res. Appl.*, vol. 16, no. 2, pp. 138–151, 2022.
- [18] A. Alberdi, A. Aztiria, A. Basarab, and D. J. Cook, “Using smart offices to predict occupational stress,” *Int. J. Ind. Ergon.*, vol. 67, pp. 13–26, 2018.
- [19] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, “Lexicon-based methods for sentiment analysis,” *Comput. Linguist. Assoc. Comput. Linguist.*, vol. 37, no. 2, pp. 267–307, 2011.
- [20] Z. Yan-Yan, Q. Bing, and L. Ting, “Integrating intra-and inter-document evidences for improving sentence sentiment classification,” *Automatica Sinica*, vol. 36, no. 10, pp. 1417–1425, 2010.
- [21] H. Yu and V. Hatzivassiloglou, “Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences,” in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, Association for Computational Linguistics, 2003, pp. 129–136.
- [22] P. D. Turney and M. L. Littman, “Measuring praise and criticism: Inference of semantic orientation from association,” *ACM Trans. Inf. Syst.*, vol. 21, no. 4, pp. 315–346, 2003.

- [23] V. Hatzivassiloglou and J. M. Wiebe, “Effects of adjective orientation and gradability on sentence subjectivity,” in *Proceedings of the 18th conference on Computational linguistics -*, 2000.
- [24] A. Moreo, M. Romero, J. L. Castro and J. M. Zurita, “Lexicon-based Comments-oriented News Sentiment Analyzer system,” *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9166–9180, 2012.
- [25] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” *Artif. Intell. Rev.*, vol. 55, no. 7, pp. 5731–5780, 2022.
- [26] I. Chetviorkin and N. V. Loukachevitch, “Extraction of Russian sentiment lexicon for product meta-domain,” *International Conference on Computational Linguistics*, 2012.
- [27] M. Kaity and V. Balakrishnan, “Sentiment lexicons and non-English languages: a survey,” *Knowl. Inf. Syst.*, vol. 62, no. 12, pp. 4445–4480, 2020.
- [28] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: <https://doi.org/10.1038/nature14539>.
- [29] S. Ruder, “An Overview of Multi-Task Learning in Deep Neural Networks,” arXiv.org, 2017. <https://arxiv.org/abs/1706.05098>
- [30] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A neural probabilistic language model,” *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.
- [31] Z. S. Harris, “Distributional Structure,” *WORD*, vol. 10, no. 2–3, pp. 146–162, 1954.
- [32] T. Mikolov, K. Chen, G. Corrado and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv [cs.CL]*, 2013.
- [33] J. Pennington, R. Socher and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [34] I. Sutskever, O. Vinyals and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2014, pp. 3104–3112.
- [35] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. London, England: MIT Press, 2016.

- [36] F. Chollet, *Deep learning with python*, 1st Edition. New York, NY: Manning Publications, 2017.
- [37] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.
- [38] A. Malte and P. Ratadiya, “Multilingual cyber abuse detection using advanced transformer architecture,” in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, IEEE, 2019, pp. 784–789.
- [39] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [40] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modelling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [41] A. Vaswani et al., “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.
- [42] M. E. Peters et al., “Deep contextualized word representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.
- [43] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A lite BERT for self-supervised learning of language representations,” *arXiv [cs.CL]*, p. arXiv:1909.11942, 2019.
- [44] Y. Liu et al., “RoBERTa: A robustly optimized BERT pretraining approach,” *arXiv [cs.CL]*, 2019.
- [45] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners,” *Cloudfront.net*. [Online]. Available: [https://d4mucfpksywv.cloudfront.net/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf](https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf). [Accessed: 11-Dec-2022].
- [46] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” in *Advances in Neural Information Processing Systems*, 2019.

- [47] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” *arXiv [cs.LG]*, 2019.
- [48] Chaudhary, “A visual survey of data augmentation in NLP,” *Amit Chaudhary*, 17-May-2020. [Online]. Available: <https://amitnss.com/2020/05/data-augmentation-for-nlp>. [Accessed: 29-Apr-2021].
- [49] R. Barzilay and K. R. McKeown, “Extracting paraphrases from a parallel corpus,” in Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01, 2001.
- [50] N. Madnani and B. J. Dorr, “Generating phrasal and sentential paraphrases: A survey of data-driven methods,” *Comput. Linguist. Assoc. Comput. Linguist.*, vol. 36, no. 3, pp. 341–387, 2010.
- [51] B. Li, Y. Hou, and W. Che, “Data augmentation approaches in natural language processing: A survey,” *AI Open*, 2022.
- [52] H. Fukuda, T. Tsunakawa, J. Oshima, R. Oshima, M. Nishida, and M. Nishimura, “BERT-based automatic text scoring for collaborative learning,” in *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, 2020.
- [53] J. Wang, H. C. Chen, and R. Radach, *Reading Chinese script: A cognitive analysis*. Psychology Press, 1999.
- [54] J. Wei and K. Zou, “EDA: Easy data augmentation techniques for boosting performance on text classification tasks,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 6382–6388.
- [55] H. Guo, Y. Mao, and R. Zhang, “Augmenting data with Mixup for sentence classification: An empirical study,” *arXiv [cs.CL]*, 2019.
- [56] B. Tahayna, R. K. Ayyasamy, N. B. A. Jalil, A. Sangodiah, L. N. Tahayna, and S. Krisnan, “Disparity-aware Pandemic Response Classification by Fine-Tuning Transfer Learning Approach,” in *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, IEEE, 2022, pp. 25–28.



- [57] A. Gelbukh, Ed., *Computational linguistics and intelligent text processing*. Cham: Springer International Publishing, 2015.
- [58] ILS Team, “Twitter Usage Statistics,” *Internet live stats*, 2020. [Online]. Available: <https://www.internetlivestats.com/twitter-statistics/>. [Accessed: 21-Dec-2021].
- [59] S. Merayo-Alba, E. Fidalgo, V. González-Castro, R. Alaiz-Rodríguez, and J. Velasco-Mata, “Use of natural language processing to identify inappropriate content in text,” in *Lecture Notes in Computer Science*, Cham: Springer International Publishing, 2019, pp. 254–263.
- [60] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, “The impact of features extraction on the sentiment analysis,” *Procedia Comput. Sci.*, vol. 152, pp. 341–348, 2019.
- [61] A. Buche, “Opinion Mining and analysis: A survey,” *Int. j. nat. lang. comput.*, vol. 2, no. 3, pp. 39–48, 2015.
- [62] W. C. F. Mariel, S. Mariyah, and S. Pramana, “Sentiment analysis: a comparison of deep learning neural network algorithm with SVM and naïve Bayes for Indonesian text,” in *Journal of Physics: Conference Series*, vol. 971, IOP Publishing, 2018.
- [63] A. Onan, “Deep learning based sentiment analysis on product reviews on twitter,” in *Communications in Computer and Information Science*, Cham: Springer International Publishing, 2019, pp. 80–91.
- [64] S. Peng *et al.*, “A survey on deep learning for textual emotion analysis in social networks,” *Digit. Commun. Netw.*, vol. 8, no. 5, pp. 745–762, 2022.
- [65] P. Zhang and Z. He, “Using data-driven feature enrichment of text representation and ensemble technique for sentence-level polarity classification,” *Journal of Information Science*, vol. 41, no. 4, pp. 531–549, 2015.
- [66] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, “Very deep convolutional networks for text classification,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017.
- [67] Y. Chen, *Convolutional neural network for sentence classification*, M.S. thesis, College of Computer Science, University of Waterloo, 2015. 2016.

- [68] X. Gong, W. Ying, S. Zhong, and S. Gong, “Text sentiment analysis based on transformer and augmentation,” *Front. Psychol.*, vol. 13, p. 906061, 2022.
- [69] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *IEEE Computational Intelligence magazine*, vol. 13, pp. 55–75, 2018.
- [70] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J. Big Data*, vol. 8, no. 1, p. 53, 2021.
- [71] S. Casola, I. Lauriola, and A. Lavelli, “Pre-trained transformers: an empirical comparison,” *Machine Learning with Applications*, vol. 9, no. 100334, p. 100334, 2022.
- [72] X. Han *et al.*, “Pre-trained models: Past, present and future,” *AI Open*, vol. 2, pp. 225–250, 2021.
- [73] K. Rudra, A. Chakraborty, N. Ganguly, and S. Ghosh, “Understanding the usage of idioms in the twitter social network,” in *Pattern Recognition and Big Data*, WORLD SCIENTIFIC, 2017, pp. 767–788.
- [74] E. Zuo, H. Zhao, B. Chen, and Q. Chen, “Context-specific heterogeneous graph convolutional network for implicit sentiment analysis,” *IEEE Access*, vol. 8, pp. 37967–37975, 2020.
- [75] B. Liu, *Sentiment Analysis and Opinion Mining*. Cham: Springer International Publishing, 2012.
- [76] M. Z. Asghar, A. Khan, A. Bibi, F. M. Kundi, and H. Ahmad, “Sentence-level emotion detection framework using rule-based classification,” *Cognit. Comput.*, vol. 9, no. 6, pp. 868–894, 2017.
- [77] W. Anwar, X. Wang, and X.-L. Wang, “A survey of automatic Urdu language processing,” in *2006 International Conference on Machine Learning and Cybernetics*, 2006, pp. 4489–4494.
- [78] M. I. Liaqat, M. Awais Hassan, M. Shoaib, S. K. Khurshid, and M. A. Shamseldin, “Sentiment analysis techniques, challenges, and opportunities: Urdu language-based analytical study,” *PeerJ Comput. Sci.*, vol. 8, no. e1032, p. e1032, 2022.

- [79] C. Wu, F. Wu, S. Wu, Z. Yuan, and Y. Huang, “A hybrid unsupervised method for aspect term and opinion target extraction,” *Knowl. Based Syst.*, vol. 148, pp. 66–73, 2018.
- [80] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, “Sentiment analysis based on deep learning: A comparative study,” *Electronics (Basel)*, vol. 9, no. 3, p. 483, 2020.
- [81] P. Ray and A. Chakrabarti, “A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis,” *Applied Computing and Informatics*, 2020.
- [82] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, “Topic sentiment mixture: modeling facets and opinions in weblogs,” in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 171–180.
- [83] C. Lin and Y. He, “Joint sentiment/topic model for sentiment analysis,” in *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*, 2009.
- [84] Y. Jo and A. H. Oh, “Aspect and sentiment unification model for online review analysis,” in *Proceedings of the fourth ACM international conference on Web search and data mining - WSDM '11*, 2011.
- [85] Y. He, C. Lin, W. Gao, and K.-F. Wong, “Dynamic joint sentiment-topic model,” *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 1–21, 2013.
- [86] P. D. Turney, “Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, 2001, pp. 417–424.
- [87] T. Zagibalov and J. Carroll, “Automatic seed word selection for unsupervised sentiment classification of Chinese text,” in *Proceedings of the 22nd International Conference on Computational Linguistics - COLING '08*, 2008.
- [88] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai, “Automatic construction of a context-aware sentiment lexicon: An optimization approach,” in *Proceedings of the 20th international conference on World wide web - WWW '11*, 2011.

- [89] S. Huang, Z. Niu, and C. Shi, “Automatic construction of domain-specific sentiment lexicon based on constrained label propagation,” *Knowl. Based Syst.*, vol. 56, pp. 191–200, 2014.
- [90] P. Sudhir and V. D. Suresh, “Comparative study of various approaches, applications and classifiers for sentiment analysis,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 205–211, 2021.
- [91] L. L. Dhande and P. G. K. Patnaik, “Analyzing sentiment of movie review data using Naive Bayes neural classifier,” *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 3, Issue 4, July-August 2014.
- [92] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, “A survey of sentiment analysis in social media,” *Knowl. Inf. Syst.*, vol. 60, no. 2, pp. 617–663, 2019.
- [93] P. Chikersal, S. Poria, and E. Cambria, “SeNTU: Sentiment analysis of tweets by combining a rule-based classifier with supervised learning,” in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 647–651.
- [94] M. T. Khan, M. Durrani, A. Ali, I. Inayat, S. Khalid, and K. H. Khan, “Sentiment analysis and the complex natural language,” *Complex Adapt. Syst. Model.*, vol. 4, no. 1, 2016.
- [95] B. Samal, A. K. Behera, and M. Panda, “Performance analysis of supervised machine learning techniques for sentiment analysis,” in *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*, IEEE, 2017, pp. 128–133.
- [96] N. A. D. Suhaimi and H. Abas, “A Systematic Literature Review on Supervised Machine Learning algorithms,” *PERINTIS eJournal*, vol. 10, no. 1, pp. 1–24, 2020.
- [97] S. K. Jain and P. Singh, “Systematic survey on sentiment analysis,” in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018, pp. 561–565.
- [98] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *Proceedings of the 25th international conference on Machine learning - ICML '08*, 2008.

- [99] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A Convolutional Neural Network for Modelling Sentences,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 655–665.
- [100] A. Severyn and A. Moschitti, “Twitter sentiment analysis with deep convolutional neural networks,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15*, 2015.
- [101] D. Stojanovski, G. Strezoski, G. Madjarov, and I. Dimitrovski, “Twitter sentiment analysis using deep convolutional neural network,” in *International Conference on Hybrid Artificial Intelligence Systems*, Cham: Springer, 2015, pp. 726–737.
- [102] D.-T. Vo and Y. Zhang, “Target-dependent twitter sentiment classification with rich automatic features,” in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 1347–1353.
- [103] A. Connelly, V. Kuri, and M. Palomino, “Lack of consensus among sentiment analysis tools: A suitability study for SME firms,” *Plymouth.ac.uk*. [Online]. Available: <https://pearl.plymouth.ac.uk/bitstream/handle/10026.1/17530/ltc-039-connelly.pdf?sequence=1>. [Accessed: 11-Dec-2022].
- [104] F. A. Pozzi, E. Fersini, and E. Messina, “Bayesian model averaging and model selection for polarity classification,” in *International conference on application of natural language to information systems*, Berlin, Heidelberg: Springer, 2013, pp. 189–200.
- [105] E. Fersini, E. Messina, and F. A. Pozzi, “Sentiment analysis: Bayesian Ensemble Learning,” *Decis. Support Syst.*, vol. 68, pp. 26–38, 2014.
- [106] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, “Sentiment classification: The contribution of ensemble learning,” *Decis. Support Syst.*, vol. 57, pp. 77–93, 2014.
- [107] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka Jr, “Tweet sentiment analysis with classifier ensembles,” *Decis. Support Syst.*, vol. 66, pp. 170–179, 2014.
- [108] Y. Lin, X. Wang, Y. Li, and A. Zhou, “Integrating the optimal classifier set for sentiment analysis,” *Soc. Netw. Anal. Min.*, vol. 5, no. 1, 2015.

- [109] P. Zhang and Z. He, “Using data-driven feature enrichment of text representation and ensemble technique for sentence-level polarity classification,” *Journal of Information Science*, vol. 41, no. 4, pp. 531–549, 2015.
- [110] S. Tan and J. Zhang, “An empirical study of sentiment analysis for chinese documents.’ *Expert Systems with applications*,” vol. 34, pp. 2622–2629, 2008.
- [111] A. Sharma and S. Dey, “Performance investigation of feature selection methods and sentiment lexicons for sentiment analysis,” *IJCA Special Issue on Advanced Computing and Communication Technologies for HPC Applications*, vol. 3, pp. 15–20, 2012.
- [112] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, “Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach,” in *Proceedings of the 20th ACM international conference on Information and knowledge management*, ACM, 2021.
- [113] R. Mukras, N. Wiratunga, R. Lothian, S. Chakraborti, and D. Harper, “Information gain feature selection for ordinal text classification using probability re-distribution,” *Proceedings of the Textlink workshop at IJCAI*, vol. 7, 2007.
- [114] A. Abbasi, H. Chen, and A. Salem, “Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums,” *ACM Trans. Inf. Syst.*, vol. 26, no. 3, pp. 1–34, 2008.
- [115] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, “Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence,” in *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, 2016.
- [116] G. Paltoglou and M. Thelwall, “A study of information retrieval weighting schemes for sentiment analysis,” in *Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010, pp. 1386–1395.
- [117] J. Martineau and T. Finin, “Delta tfidf: An improved feature space for sentiment analysis,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 3, pp. 258–261, 2009.

- [118] B. Pang, L. Lee, and S. Vaithyanathan, ‘Thumbs up? Sentiment classification using machine learning techniques. 2002.
- [119] T. Mullen and N. Collier, “‘Sentiment analysis using support vector machines with diverse information sources,” in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 412–418.
- [120] C. Whitelaw, N. Garg, and S. Argamon, “Using appraisal groups for sentiment analysis,” in Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM ’05, 2005.
- [121] R. Xia and C. Zong, “‘A POS-based ensemble model for cross-domain sentiment classification,” in *Proceedings of 5th international joint conference on natural language processing*, 2011, pp. 614–622.
- [122] E. Riloff, S. Patwardhan, and J. Wiebe, “Feature subsumption for opinion analysis,” in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing - EMNLP ’06, 2006.
- [123] A. Kennedy and D. Inkpen, “SENTIMENT CLASSIFICATION of MOVIE REVIEWS USING CONTEXTUAL VALENCE SHIFTERS,” *Comput. Intell.*, vol. 22, no. 2, pp. 110–125, 2006.
- [124] G. Tsatsaronis and V. Panagiotopoulou, “‘A generalized vector space model for text retrieval based on semantic relatedness,” in *Proceedings of the Student Research Workshop at EACL 2009*, 2009, pp. 70–78.
- [125] S. Sani, N. Wiratunga, S. Massie, and R. Lothian, “‘Supervised Semantic Indexing Using Sub-spacing,” in *International Conference on Case-Based Reasoning*, Cham: Springer, 2014, pp. 420–434.
- [126] N. Wiratunga, I. Koychev, and S. Massie, “Feature selection and generalisation for retrieval of textual cases,” in *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 806–820.
- [127] B. Mccann, J. Bradbury, C. Xiong, and R. Socher, Learned in translation: Contextualized word vectors.’ *Advances in neural information processing systems*. 2017.
- [128] M. Ott, S. Edunov, and D. Grangier, “‘Scaling neural machine translation,” in *Proceedings of the Third Conference on Machine*

- Translation: Research Papers*, Belgium, Brussels: Association for Computational Linguistics, 2018, pp. 1–9.
- [129] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, “Unsupervised Data Augmentation for Consistency Training,” *arXiv [cs.LG]*, 2019.
- [130] B. Liu, *Sentiment Analysis and Opinion Mining*. Cham: Springer International Publishing, 2012.
- [131] S. Kannan *et al.*, “Big Data Analytics for Social Media,” in *Big Data*, Elsevier, 2016, pp. 63–94.
- [132] N. Mukhtar, M. A. Khan, and N. Chiragh, “Lexicon-based approach outperforms Supervised Machine Learning approach for Urdu Sentiment Analysis in multiple domains,” *Telemat. inform.*, vol. 35, no. 8, pp. 2173–2183, 2018.
- [133] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, “Lexicon-based methods for sentiment analysis,” *Comput. Linguist. Assoc. Comput. Linguist.*, vol. 37, no. 2, pp. 267–307, 2011.
- [134] A. Malik, Y. T. Javeri, M. Shah, and R. Mangrulkar, “Impact analysis of COVID-19 news headlines on global economy,” in *Cyber-Physical Systems*, Academic Press, 2022, pp. 189–206.
- [135] K. Denecke, “Using SentiWordNet for multilingual sentiment analysis,” in *2008 IEEE 24th International Conference on Data Engineering Workshop*, 2008.
- [136] K. Dashtipour *et al.*, “Multilingual sentiment analysis: State of the art and independent comparison of techniques,” *Cognit. Comput.*, vol. 8, no. 4, pp. 757–771, 2016.
- [137] M. Kaity and V. Balakrishnan, “Sentiment lexicons and non-english languages: a survey,” *Knowledge and Information Systems*, vol. 62, no. 12, pp. 4445–4480.
- [138] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis,” *Comput. Linguist. Assoc. Comput. Linguist.*, vol. 35, no. 3, pp. 399–433, 2009.
- [139] C. Hutto and E. Gilbert, “VADER: A parsimonious rule-based model for sentiment analysis of social media text,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, pp. 216–225, 2014.



- [140] S. Baccianella, A. Esuli, and F. Sebastiani, “SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining,” in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*, 2010.
- [141] B. Ohana and B. Tierney, “Sentiment classification of reviews using SentiWordNet,” *Proceedings of IT&T*, vol. 8, 2009.
- [142] U. Khan *et al.*, “A review of Urdu sentiment analysis with multilingual perspective: A case of Urdu and Roman Urdu language,” *Computers*, vol. 11, no. 1, p. 3, 2021.
- [143] S. Almatarneh and P. Gamallo, “A lexicon based method to search for extreme opinions,” *PLoS One*, vol. 13, no. 5, p. e0197816, 2018.
- [144] E. Cambria, D. Olsher, and D. Rajagopal, “SenticNet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis,” *Proc. Conf. AAAI Artif. Intell.*, vol. 28, no. 1, pp. 1515–1521, 2014.
- [145] N. Gupta and R. Agrawal, “Application and techniques of opinion mining,” in *Hybrid Computational Intelligence*, Elsevier, 2020, pp. 1–23.
- [146] L. Wu, F. Morstatter, and H. Liu, “SlangSD: building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification,” *Lang. Resour. Eval.*, vol. 52, no. 3, pp. 839–852, 2018.
- [147] S. M. Mohammad and P. D. Turney, “Crowdsourcing a word-emotion association lexicon,” *Computational intelligence*, vol. 29, no. 3, pp. 436–465, 2013.
- [148] L. Williams, C. Bannister, M. Arribas-Ayllon, A. Preece, and I. Spasić, “The role of idioms in sentiment analysis,” *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7375–7385, 2015.
- [149] Y. Hong, H. Kwak, Y. Baek, and S. Moon, “Tower of babel: A crowdsourcing game building sentiment lexicons for resource-scarce languages,” in *WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 549–556.
- [150] M. Lafourcade, N. Le Brun, and A. Joubert, “Collecting and evaluating lexical polarity with a Game With A Purpose,” *Aclanthology.org*. [Online]. Available: <https://aclanthology.org/R15-1044.pdf>. [Accessed: 11-Dec-2022].

- [151] M. Abdul-Mageed, M. Diab, and S. Kübler, “SAMAR: Subjectivity and sentiment analysis for Arabic social media,” *Comput. Speech Lang.*, vol. 28, no. 1, pp. 20–37, 2014.
- [152] K. Trakultaweekoon and S. Klaithin, “SenseTag: A tagging tool for constructing Thai sentiment lexicon,” in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2016.
- [153] X. Ding, B. Liu, and P. S. Yu, “A holistic lexicon-based approach to opinion mining,” in *Proceedings of the international conference on Web search and web data mining - WSDM '08*, 2008.
- [154] K. Shudo and T. Tanabe, “JDMWE: A Japanese dictionary of multi-word expressions,” *J. Nat. Lang. Process.*, vol. 17, no. 5, pp. 51–74, 2010.
- [155] A. Mudinas, D. Zhang, and M. Levene, “Combining lexicon and learning based approaches for concept-level sentiment analysis,” in *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM '12*, 2012.
- [156] S.-X. Xie and T. Wang, “Construction of unsupervised sentiment classifier on idioms resources,” *J. Cent. S. Univ.*, vol. 21, no. 4, pp. 1376–1384, 2014.
- [157] H. S. Ibrahim, S. M. Abdou, and M. Gheith, “Idioms-proverbs lexicon for modern standard Arabic and colloquial sentiment analysis,” *arXiv [cs.CL]*, 2015.
- [158] I. Spasic, L. Williams, and A. Buerki, “Idiom-based features in sentiment analysis: Cutting the Gordian knot,” *IEEE Trans. Affect. Comput.*, vol. 11, no. 2, pp. 189–199, 2020.
- [159] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge, England: Cambridge University Press, 2015.
- [160] X. Chen, Z. Hai, S. Wang, D. Li, C. Wang, and H. Luan, “Metaphor identification: A contextual inconsistency based neural sequence labeling approach,” *Neurocomputing*, vol. 428, pp. 268–279, 2021.
- [161] K. Matsumoto, S. Tsuchiya, M. Yoshida, and K. Kita, “Construction and Expansion of Dictionary of Idiomatic Emotional Expressions and Idiomatic Emotional Expression Corpus,” *International Journal of Computer & Software Engineering*, vol. 6, no. 2, 2021.

- [162] J. Briskilal and C. N. Subalalitha, “An ensemble model for classifying idioms and literal texts using BERT and RoBERTa,” *Inf. Process. Manag.*, vol. 59, no. 1, p. 102756, 2022.
- [163] K. Dashtipour, M. Gogate, A. Gelbukh, and A. Hussain, “Extending persian sentiment lexicon with idiomatic expressions for sentiment analysis,” *Soc. Netw. Anal. Min.*, vol. 12, no. 1, 2022.
- [164] A. Hwang and C. Hidey, “Confirming the non-compositionality of idioms for sentiment analysis,” in *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, 2019, pp. 125–129.
- [165] J. Lin, D. Campos, N. Craswell, B. Mitra, and E. Yilmaz, “Significant improvements over the state of the art? A case study of the MS MARCO document ranking leaderboard,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
- [166] J. Montantes, “BERT Transformers — how do they work? - becoming human: Artificial intelligence magazine,” *Becoming Human: Artificial Intelligence Magazine*, 12-Apr-2021. [Online]. Available: <https://becominghuman.ai/bert-transformers-how-do-they-work-cd44e8e31359>. [Accessed: 11-Dec-2022].
- [167] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *arXiv [cs.CL]*, 2019.
- [168] Y. Wu et al., “Google’s Neural Machine Translation system: Bridging the gap between human and Machine Translation,” *arXiv [cs.CL]*, 2016.
- [169] T. McCoy, E. Pavlick, and T. Linzen, “Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3428–3448.
- [170] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith, “Annotation artifacts in natural language inference data,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 107–112.
- [171] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme, “Hypothesis only baselines in natural language inference,” in *Proceedings*

- of the Seventh Joint Conference on Lexical and Computational Semantics, 2018.
- [172] Y. Zhu *et al.*, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [173] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.
- [174] D. Araci, “FinBERT: Financial sentiment analysis with pre-trained language models,” *arXiv [cs.CL]*, 2019.
- [175] S. M. Rezaeina, R. Rahmani, A. Ghodsi, and H. Veisi, “Sentiment analysis based on improved pre-trained word embeddings,” *Systems with Applications*, vol. 117, pp. 139–147, 2019.
- [176] D. R. Beddiar, M. S. Jahan, and M. Oussalah, “Data expansion using back translation and paraphrasing for hate speech detection,” *Online Soc. Netw. Media*, vol. 24, no. 100153, p. 100153, 2021.
- [177] W. Wang, B. Li, D. Feng, A. Zhang, and S. Wan, “The OL-DAWE model: Tweet polarity sentiment analysis with data augmentation,” *IEEE Access*, vol. 8, pp. 40118–40128, 2020.
- [178] C. Fellbaum. *WordNet: An electronic lexical resource*, in *The Oxford Handbook of Cognitive Science*, 301–314. Oxford: Oxford University Press.
- [179] S. Edunov, M. Ott, M. Auli, and D. Grangier, “Understanding back-translation at scale,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 489–500.
- [180] S. Edunov, M. Ott, M. Ranzato, and M. Auli, “On the evaluation of machine translation systems trained with back-translation,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2836–2846.
- [181] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” *arXiv [cs.LG]*, pp. 649–657, 2015.
- [182] W. Y. Wang and D. Yang, “That’s so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets,”

- in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2557–2563.
- [183] G. Nicolai, B. Hauer, A. St Arnaud, and G. Kondrak, “Morphological reinflection via discriminative string transduction,” in *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2016, pp. 31–35.
- [184] W. Yu *et al.*, “QANet: Combining Local Convolution with Global SelfAttention for Reading Comprehension,” in *International Conference on Learning Representations*, 2018.
- [185] J. Zhang and T. Matsumoto, “Corpus augmentation for neural machine translation with Chinese-Japanese parallel corpora,” *Applied sciences*, vol. 9, no. 10, 2019.
- [186] S. Kobayashi, “Contextual augmentation: Data augmentation by words with paradigmatic relations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 452–457.
- [187] G. Rizos, K. Hemker, and B. Schuller, “Augment to prevent: short-text data augmentation in deep learning for hate-speech classification,” in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 991–1000.
- [188] P. Vijayaraghavan, S. Vosoughi, and D. Roy, “Automatic detection and categorization of election-related tweets,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 10, no. 1, pp. 703–706, 2021.
- [189] F. D. Souza and J. B. D. O. Souza, “Sentiment Analysis on Brazilian Portuguese User Reviews,” in *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, IEEE, 2021, pp. 1–6.
- [190] Y. Yaghoobzadeh, K. Kann, T. J. Hazen, E. Agirre, and H. Schütze, “Probing for semantic classes: Diagnosing the meaning content of word embeddings,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5740–5753.

- [191] G. G. Şahin, “To augment or not to augment? A comparative study on text augmentation techniques for low-resource NLP,” *Comput. Linguist. Assoc. Comput. Linguist.*, vol. 48, no. 1, pp. 5–42, 2022.
- [192] C.E. Osgood, G.J. Suci and P.H. Tannenbaum, *The measurement of meaning*. University of Illinois Press, Urbana.
- [193] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Found. Trends® Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, 2008.
- [194] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014.
- [195] L. Williams, “Pushing the envelope of sentiment analysis beyond words and polarities,” Cardiff University, 2017.
- [196] K. Krippendorff, “Content analysis,” *SAGE Publications Inc*, 08-Dec-2022. [Online]. Available: <https://us.sagepub.com/en-us/nam/content-analysis/book258450>. [Accessed: 11-Dec-2022].
- [197] K. Krippendorff, “Reliability in content analysis: Some common misconceptions and recommendations,” *Hum. Commun. Res.*, vol. 30, no. 3, pp. 411–433, 2004.
- [198] J. Geertzen, “Inter-Rater Agreement with multiple raters and variables.” [Online]. Available: <https://mlnl.net/jg/software/ira/>. [Accessed: 20-Feb-2022].
- [199] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.
- [200] Q. Yang, “Sentiment Analysis of Covid-19 related Tweets. Kaggle,” *Kaggle*, 2021. [Online]. Available: <https://kaggle.com/competitions/sentiment-analysis-of-covid-19-related-tweets>. [Accessed: 12-Sep-2021].
- [201] S. J. Russell and P. Norvig, “Artificial intelligence: a modern approach,” Pearson Education, 2010.
- [202] Ł. Augustyniak, P. Szymański, T. Kajdanowicz, and W. Tuligłowicz, “Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis,” *Entropy*, vol. 18, no. 1, p. 4, Dec. 2015, doi: <https://doi.org/10.3390/e18010004>.

- [203] Cambridge: Cambridge University Press.  
<https://dictionary.cambridge.org/dictionary/english>
- [204] J. Tiedemann, "Parallel data in OPUS--web harvesting and linguistic preprocessing," In LREC (pp. 794-801), 2012.
- [205] "Helsinki-NLP/opus-mt-de-en · Hugging Face," *huggingface.co*.  
<https://huggingface.co/Helsinki-NLP/opus-mt-de-en> (accessed Mar. 02, 2022).
- [206] "cardiffnlp/twitter-roberta-base-sentiment · Hugging Face,"  
*huggingface.co*. <https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment>
- [207] A. Go, R. Bhayani, & L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Report, Stanford, 1(12), 2009.