



Implicit Aspect Analysis Approach for Sentiment Classification

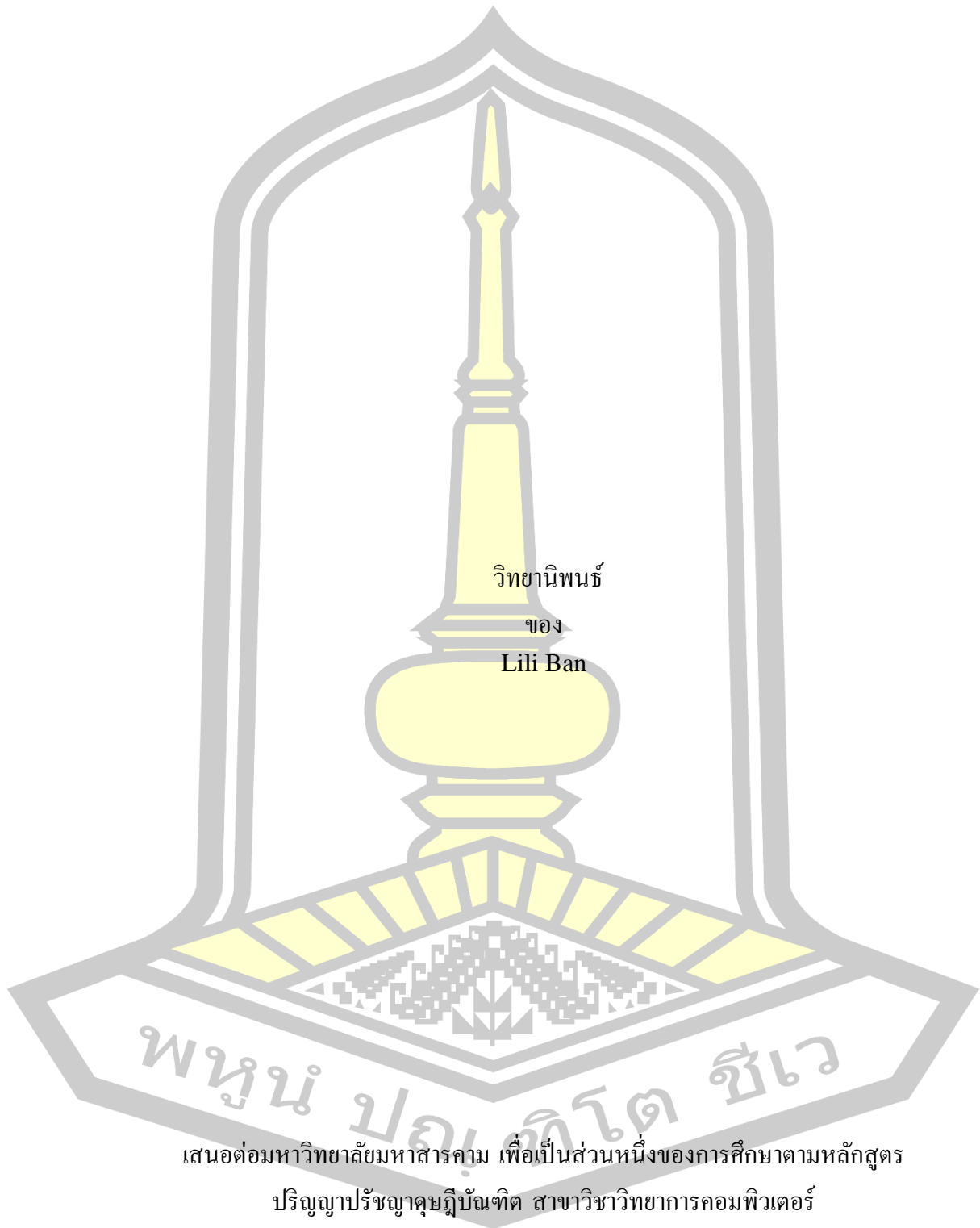
Lili Ban

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Computer Science

May 2025

Copyright of Mahasarakham University

Implicit Aspect Analysis Approach for Sentiment Classification



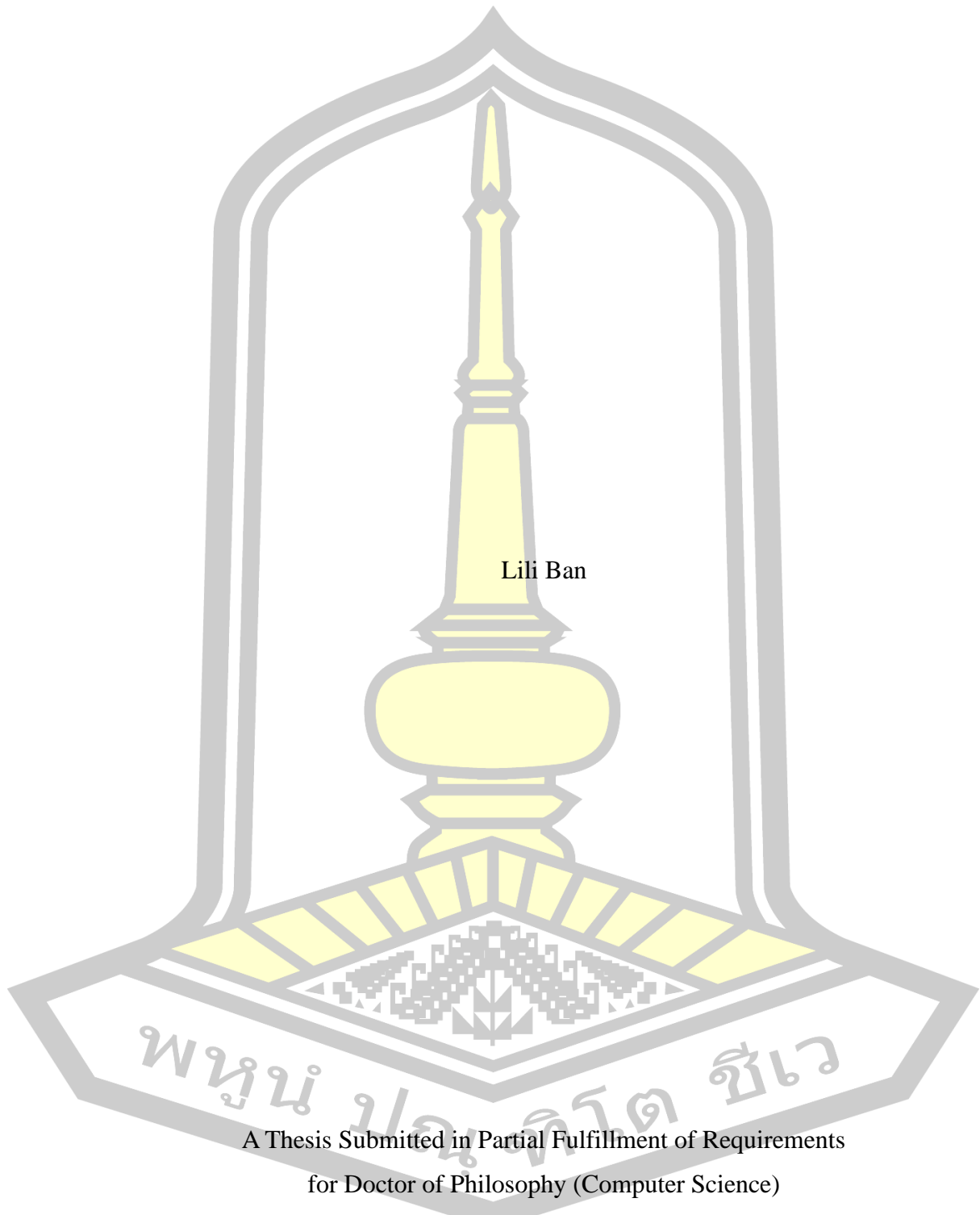
เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

พฤษภาคม 2568

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

Implicit Aspect Analysis Approach for Sentiment Classification



Lili Ban

A Thesis Submitted in Partial Fulfillment of Requirements
for Doctor of Philosophy (Computer Science)

May 2025

Copyright of Mahasarakham University



The examining committee has unanimously approved this Thesis, submitted by Ms. Lili Ban , as a partial fulfillment of the requirements for the Doctor of Philosophy Computer Science at Maharakham University

Examining Committee

Chairman

(Asst. Prof. Manasawee
Kaenampornpan , Ph.D.)

Advisor

(Assoc. Prof. Jantima Polpinij ,
Ph.D.)

Co-advisor

(Asst. Prof. Gangarn Sompasertsri ,
Ph.D.)

Committee

(Assoc. Prof. Panida Songram ,
Ph.D.)

Committee

(Bancha Luaphol , Ph.D.)

Maharakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Computer Science

(Assoc. Prof. Jantima Polpinij , Ph.D.)
Dean of The Faculty of Informatics

(Asst. Prof. Pondej Chaowarat , Ph.D.)
Dean of Graduate School

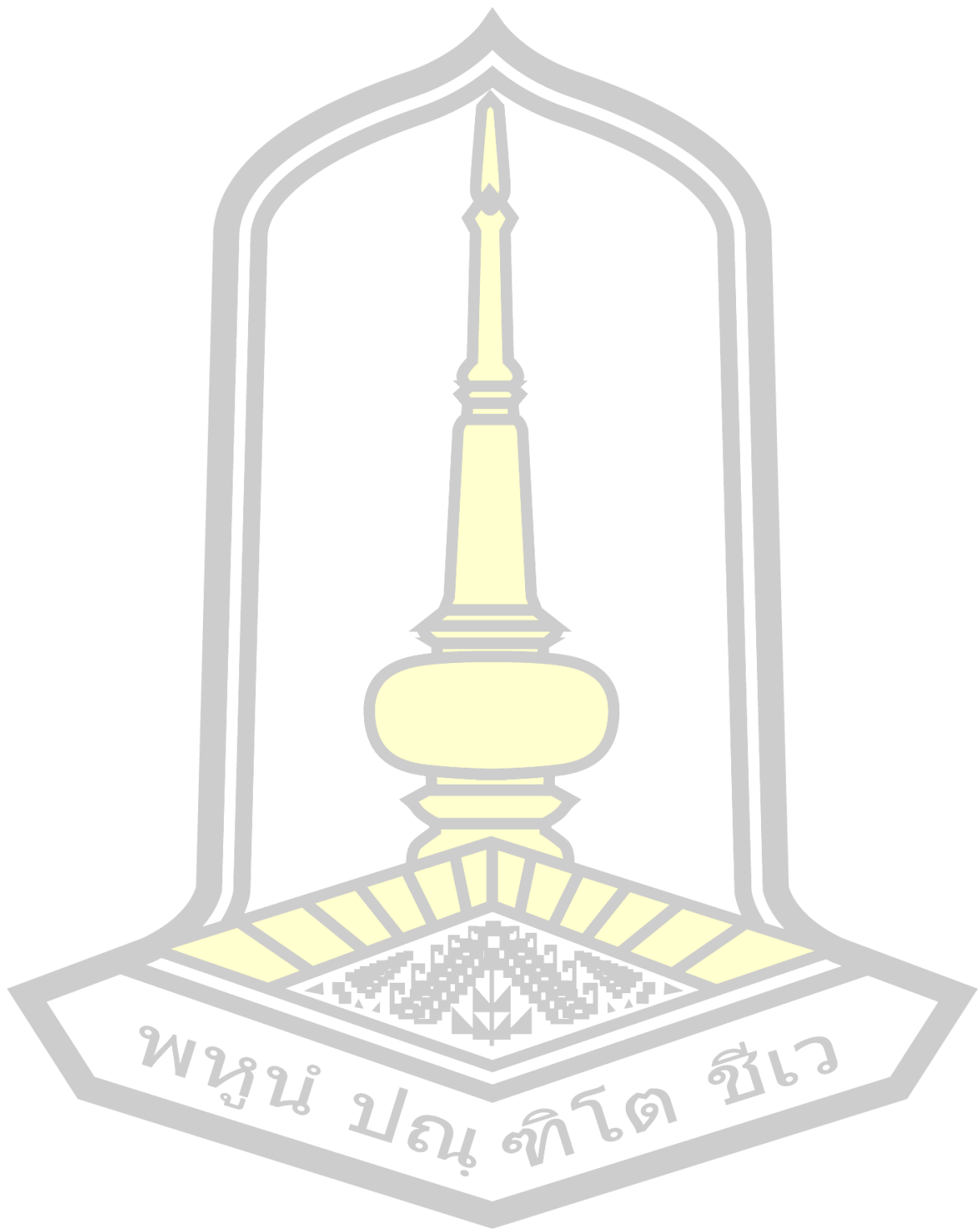
มหาวิทยาลัยราชภัฏรำไพพรรณี

TITLE	Implicit Aspect Analysis Approach for Sentiment Classification		
AUTHOR	Lili Ban		
ADVISORS	Associate Professor Jantima Polpinij , Ph.D. Assistant Professor Gamgarn Sompasertsri , Ph.D.		
DEGREE	Doctor of Philosophy	MAJOR	Computer Science
UNIVERSITY	Maharakham University	YEAR	2025

ABSTRACT

Implicit aspect-based sentiment analysis (ABSA) is a challenging task that involves identifying sentiments toward aspects that are not explicitly mentioned in text. Unlike explicit ABSA, where aspect terms are directly stated, implicit ABSA requires contextual and linguistic understanding to infer the hidden relationships between words and their associated aspects. This complexity makes it a crucial yet underexplored area in sentiment analysis. This study aims to develop an effective framework for implicit aspect identification and sentiment classification, focusing on hotel reviews as a case study. To achieve this, we propose a two-stage approach: (1) constructing an implicit aspect corpus using Word2Vec Skip-gram and dependency parsing, and (2) developing a binary aspect-based sentiment classifier that associates sentiments with specific hotel aspects, including staff service, cleanliness, value for money, and location convenience. The dataset, sourced from TripAdvisor, consists of 2,000 English-written hotel reviews, carefully annotated by linguistic experts to ensure high-quality aspect and sentiment labeling. The feature representation combines TF-IDF and Word2Vec embeddings, capturing both statistical and semantic relationships within the text. The experimental results demonstrate that deep learning models significantly outperform traditional machine learning methods in implicit aspect identification and sentiment classification. Among all models, LSTM achieved the highest accuracy (88.2%) and F1-score (90.4%) for implicit aspect identification, while SVM with a Linear Kernel achieved the best sentiment classification accuracy (93.0%). A comparative analysis with existing studies highlights that our method achieves higher accuracy and better generalization than previous approaches that relied on dictionary-based, rule-based, and probabilistic techniques. Unlike earlier models, which were often domain-dependent, our approach effectively captures hidden aspect relationships using deep learning techniques. These findings validate the effectiveness of deep learning in implicit aspect-based sentiment analysis, providing a scalable and adaptable solution for sentiment classification tasks. The study contributes to advancing sentiment analysis, implicit aspect detection, and opinion mining, making it highly valuable for businesses and researchers analyzing customer feedback.

Keyword : Implicit Aspect-Based Sentiment Analysis, Sentiment Classification, Word2Vec Skip-gram, Deep Learning, Aspect Identification, Hotel Reviews



ACKNOWLEDGEMENTS

The successful completion of this thesis would not have been possible without the support, guidance, and encouragement of many individuals and institutions, to whom I am deeply grateful.

First and foremost, I sincerely appreciate my supervisors, Associate Professor Dr. Jantima Polpinij and Associate Professor Dr. Gamgarn Somprasertsri. Your expertise, unwavering support, and insightful guidance have been invaluable throughout my academic journey. Your dedication to research, meticulous approach, and passion for knowledge have greatly influenced my growth as a researcher. I am also profoundly grateful to Dr. Lien Thai Son for his invaluable advice and guidance, which have significantly shaped my research perspective and academic development.

Throughout my Ph.D. studies, I have been fortunate to receive the support of colleagues, mentors, and friends within and beyond the Faculty of Informatics. Their generosity, encouragement, and shared experiences have enriched my journey, fostering both academic and personal growth. I am deeply appreciative of the instructors who have imparted knowledge and introduced me to innovative learning techniques.

My deepest gratitude goes to my family, whose unwavering love and belief in me have provided the foundation for my academic pursuits. I would also like to extend my sincere thanks to the Faculty of Informatics at Mahasarakham University for providing essential resources and facilities that have been instrumental in conducting this research.

I am deeply grateful to everyone I've met along the way. Your support and encouragement mean the world to me.

Lili Ban

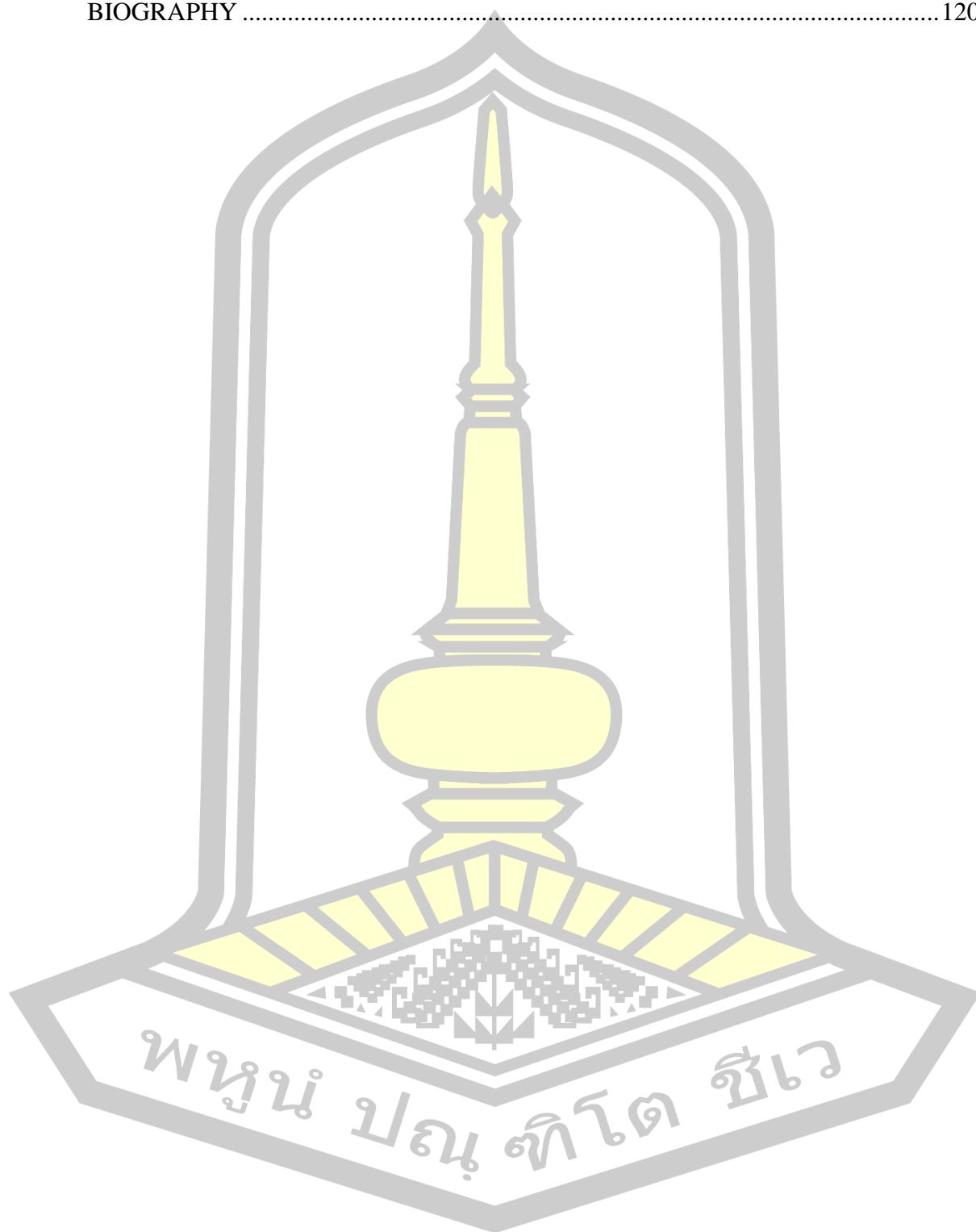
TABLE OF CONTENTS

	Page
ABSTRACT.....	D
ACKNOWLEDGEMENTS.....	F
TABLE OF CONTENTS.....	G
LIST OF TABLES.....	J
LIST OF FIGURES.....	K
CHAPTER I INTRODUCTION.....	1
1.1 Background.....	1
1.2 Research Question.....	2
1.3 Contribution of Research.....	3
1.4 Objectives of Research.....	3
1.5 Scope of Research.....	3
1.6 Research Significance.....	4
1.7 Terminologies.....	4
CHAPTER II LITERATURE REVIEWS.....	6
2.1 Customer Reviews.....	6
2.1.1 Definition.....	6
2.1.2 Benefits of Customer Review.....	6
2.1.3 Hotel Customer Reviews.....	7
2.2 Sentiment Classification.....	9
2.2.1 Definition.....	9
2.2.2 Types of Sentiment Classification.....	10
2.3 Aspect-based Text Classification (ABTC).....	12
2.3.1 Challenges and Considerations.....	12
2.3.2 Generic Framework for Aspect-based Text Classification.....	14
2.3.3 Pre-processing for Aspect-based Text Classification.....	17

2.3.4 Classifier Modeling for Aspect-based Text Classification	18
2.3.5 Evaluation Metrics for Aspect-based Text Classification	39
2.4 Word2Vec.....	43
2.5 K-means Clustering Approach.....	49
2.6 Related Work	50
CHAPTER III RESEARCH METHODOLOGY	55
3.1 Datasets.....	55
3.2 The Framework.....	57
3.3 Preliminary: Developing Implicit Word Corpus of Each Hotel Aspect.....	58
3.4 Binary Aspect-based Sentiment Classifier Development	64
3.4.1 Text Pre-processing and Representation	64
3.4.2 Classifier Modeling	69
3.4.3 Evaluation Metrics and Methodology	73
CHAPTER IV EXPERIMENTAL RESULTS AND DISCUSSION	76
4.1 Experimental Setup.....	76
4.1.1 Datasets	76
4.1.2 Data Preprocessing	81
4.1.3 Feature Representation	83
4.1.4 Evaluation Metrics	86
4.2 Experimental Results	88
4.2.1 Implicit Aspect Corpus Construction Results	88
4.2.2 Model Performance Evaluation.....	89
4.3 Comparison with Existing Studies.....	97
4.3.1 Comparison of Existing Studies on Implicit Aspect Extraction.....	98
4.3.2 Comparison of Existing Studies on Aspect-Based Sentiment Analysis...99	
CHAPTER V CONCLUSION AND FUTURE WORK	103
5.1 Conclusion	103
5.2 Challenges and Limitations of This Research	106
5.3 Future Work	107

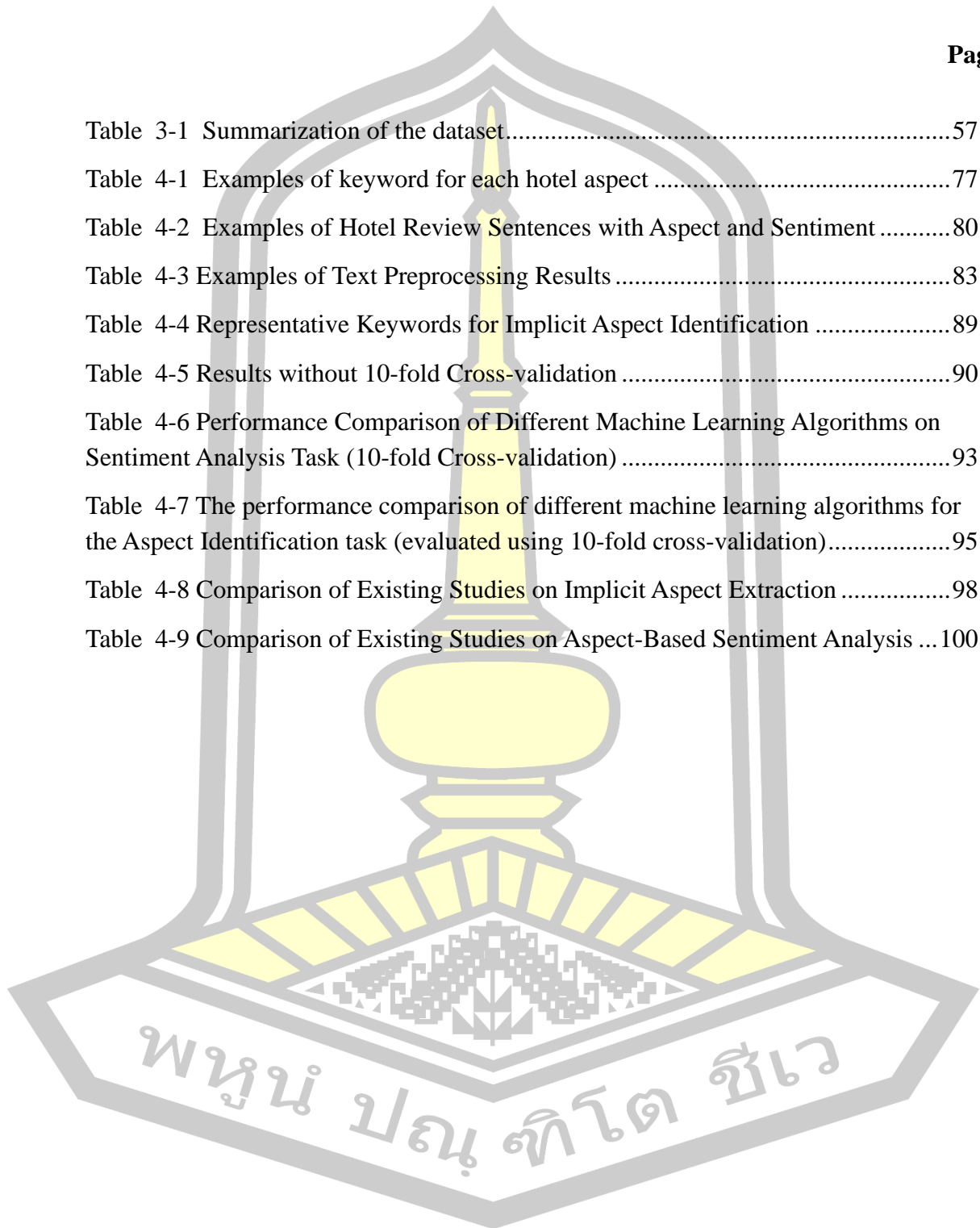
REFERENCES 108

BIOGRAPHY 120



LIST OF TABLES

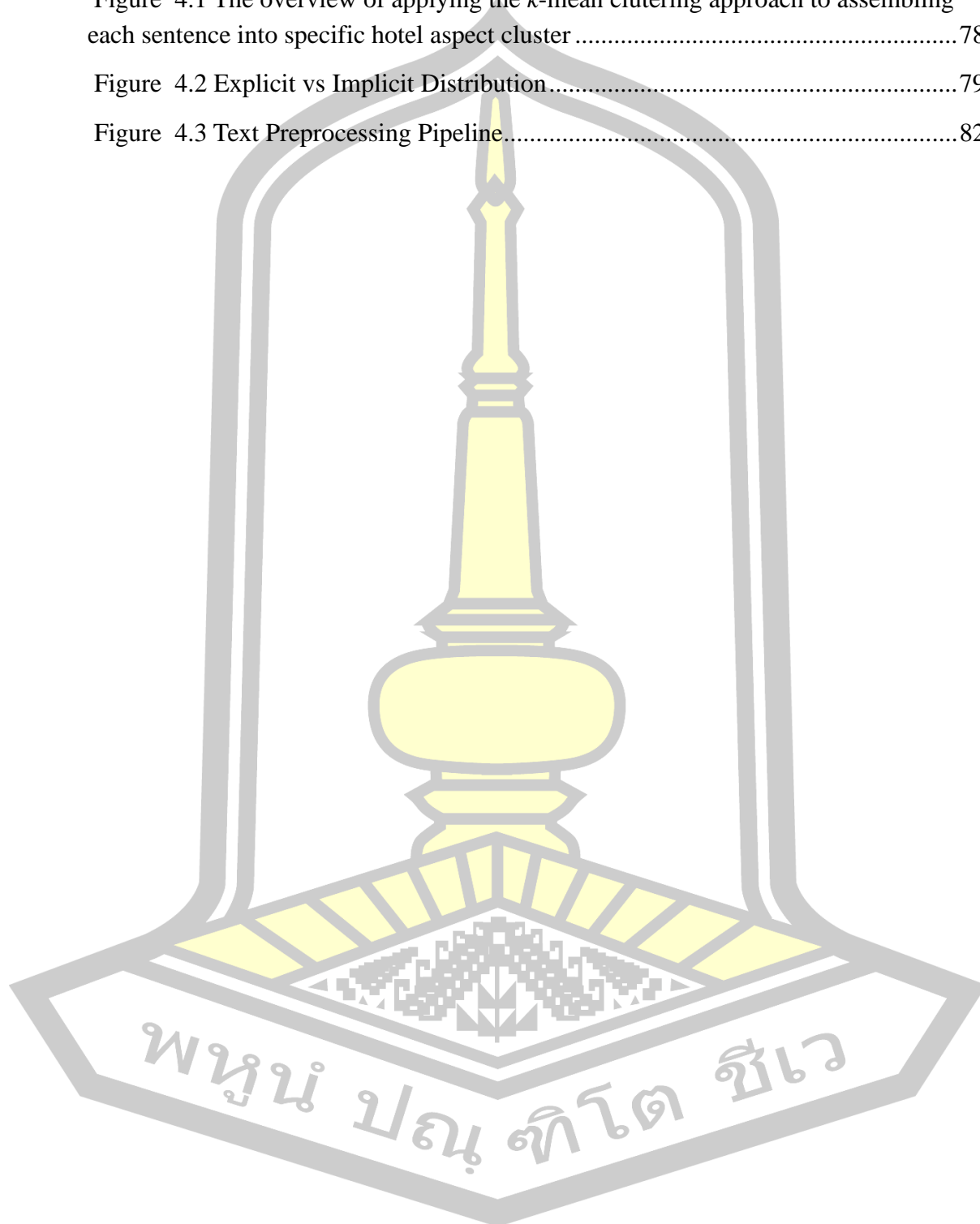
	Page
Table 3-1 Summarization of the dataset.....	57
Table 4-1 Examples of keyword for each hotel aspect	77
Table 4-2 Examples of Hotel Review Sentences with Aspect and Sentiment	80
Table 4-3 Examples of Text Preprocessing Results	83
Table 4-4 Representative Keywords for Implicit Aspect Identification	89
Table 4-5 Results without 10-fold Cross-validation	90
Table 4-6 Performance Comparison of Different Machine Learning Algorithms on Sentiment Analysis Task (10-fold Cross-validation)	93
Table 4-7 The performance comparison of different machine learning algorithms for the Aspect Identification task (evaluated using 10-fold cross-validation).....	95
Table 4-8 Comparison of Existing Studies on Implicit Aspect Extraction	98
Table 4-9 Comparison of Existing Studies on Aspect-Based Sentiment Analysis ...	100



LIST OF FIGURES

	Page
Figure 2.1 The overview of text classification.....	10
Figure 2.2 The overview of document-based text classification	11
Figure 2.3 The overview of Sentence-based text classification.....	11
Figure 2.4 The overview of aspect-based text classification	12
Figure 2.5 generic framework of aspect-based text classification.....	14
Figure 2.6 An overview of CNN structure for text classification.....	26
Figure 2.7 An overview of RNN structure for text classification.....	29
Figure 2.8 An overview of LSTM structure for text classification.....	30
Figure 2.9 An overview of GRU structure for text classification.....	30
Figure 2.10 An overview of bagging method	31
Figure 2.11 An overview of Boosting method.....	32
Figure 2.12 An overview of Stacking method	33
Figure 2.13 An overview of transformer learning for text classification.....	35
Figure 2.14 An overview of BERT for text classification.....	37
Figure 2.15 Confusion Matrix.....	39
Figure 2.16 ROC and AUC.....	42
Figure 2.17 The architecture diagram for CBOW and Skip-gram.....	44
Figure 2.18 An example of using the CBOW model to predict the word “fox”.....	46
Figure 3.1 The Framework	57
Figure 3.2 Sequential Steps in Text Pre-processing Framework	60
Figure 3.3 kip-gram Model Architecture	62
Figure 3.4 Instructions for downloading NLTK by pip	66
Figure 3.5 Feature Representation Framework.....	67
Figure 3.6 K-Fold Cross-Validation Process (k=10)	70
Figure 3.7 Overview of Machine Learning Algorithms for Sentiment Classification	73

Figure 3.8 Evaluation Framework	75
Figure 4.1 The overview of applying the <i>k</i> -mean clustering approach to assembling each sentence into specific hotel aspect cluster	78
Figure 4.2 Explicit vs Implicit Distribution.....	79
Figure 4.3 Text Preprocessing Pipeline.....	82



CHAPTER I

INTRODUCTION

1.1 Background

Sentiment analysis (SA) is the computational analysis of opinions, sentiments, and subjectivity of text [1], while aspect-based sentiment analysis (ABSA) is a specific SA that aims to extract most important aspects of an entity and predict the polarity of each aspect from the text [2]. SA on review data can be extremely advantageous for both producers and consumers. However, in review datasets, ABSA is preferred over providing an overall sentiment. This is due to the fact that it is more precise and biased towards the aspect in hand.

To the best of our knowledge, ABSA includes a number of subtasks, two of which are aspect-category detection (ACD) and aspect-category sentiment analysis (ACSA)[3]. The ACD task in sentiment analysis aims to identify the aspect categories mentioned in a sentence [3]. Meanwhile, the ACSA task aims to identify the sentiment of predefined aspect categories [4]. For example, consider following sentence, “The restaurant was expensive, but the menu was great.” The ACD component detects the aspect categories (i.e. price and food), and the ACSA component predicts the sentiment polarities toward each aspect category (i.e. price: negative and food: positive).

In fact, an “aspect” can be defined as an item’s attribute, characteristic, or feature. There are two types of aspects: implicit and explicit [5]. For example, consider following sentence, “The cost of the car is very expensive, but the car is beautiful.” This example shows that “cost” is the aspect for “car” and the polarity is positive towards the car. The “cost” is considered as an explicit aspect and “beautiful” is considered as an implicit aspect [6]. Another example can be presented following. Consider the sentence, “Lovely display, I wish to buy one despite its slight heaviness.”

The explicit aspects that is extracted from this sentence should be “display”. Meanwhile, the implicit aspect that is extracted from this sentence should be “weight” because the word “heaviness” implies to its weight [7].

It is well-known that if analyzing customer review datasets based on ABSA, the identification of an aspect should be performed as a crucial step. Unfortunately, if considering the example above, this demonstrates that it is not easy to extract implicit and explicit aspects and this issue has been confirmed by [8][9], especially extracting of implicit aspects. However, extraction of explicit aspect has been extensively studied and many approaches are proposed for it. But inadequate work has been done or extraction of implicit aspects. The extraction of explicit aspects has been extensively researched, and numerous ways have been presented. However, insufficient work has been done to extract implicit aspects [10][11]. Extraction of implicit aspects is challenging but necessary because implicit aspects are frequently encountered in opinionated documents [11][12].

Based the previous study [10][12], they have mentioned that approximately 30% of reviews lack evident opinion words but nonetheless convey clear human-aware sentiment orientation, a phenomenon known as implicit sentiment.

As mentioned above, studying for proposing of an appropriate approach for implicit aspect analysis in ABSA becomes the primary challenge in this study.

1.2 Research Question

Analyzing implicit aspects for sentiment classification involves identifying and interpreting the underlying opinions or sentiments expressed in a text that may not be directly stated. This task is more nuanced than detecting explicit expressions of sentiment because it requires understanding context, subtleties of language, and sometimes even cultural or situational knowledge. What is the approach used for analyzing implicit aspects of sentiment classification?

1.3 Contribution of Research

Word2Vec, particularly its Skip-gram model, can indeed be applied to collect words related to specific aspects of hotels, such as amenities, service quality, room conditions, location, and price, among others. The Skip-gram model is designed to predict the context words (surrounding words) for a given target word within a sentence or document. This capability makes it particularly useful for extracting words that are implicitly related to a given topic, like various hotel aspects in this case.

1.4 Objectives of Research

This study aims to present an approach of implicit aspect analysis approach for sentiment classification. The case study is hotel reviews.

1.5 Scope of Research

This is to propose a method of implicit aspect analysis for binary sentiment classification.

The dataset is related to hotel reviews downloaded from the Tripadvisor website. The hotel reviews are in English. Two or three linguistic experts may be required for two main tasks:

- 2.1 They help to develop a ground-truth of hotel reviews.
- 2.2 They help with generating implicit sentences in the test set while explaining the analysis results as required.

Four main explicit hotel aspects (features) are considered. They are staff service, cleanliness, value price, and convenience of location.

The Skip-gram model of Word2Vec is utilized to collect words that are associated with each aspect of the hotel from a combination of hotel review datasets that download from multiple websites.

Multiple machine learning algorithms, e.g. Support Vector Machines (SVM), random forest, Convolutional Neural Network (CNN), that are used to develop classifier models are compared.

Evaluation metrics used to measure the performance of the proposed classifier models could be accuracy, F1, and AUC.

1.6 Research Significance

The implicit aspect of the analysis method for sentiment classification is crucial in enhancing our understanding of sentiments expressed in text, resulting in more precise and perceptive sentiment analysis. It has the capability to be used in various applications, such as improving customer experience to enhancing natural language understanding technologies. Here are some of the main implications of this approach:

Comprehensive Understanding of Sentiment: It helps in capturing the underlying sentiment of aspects that are not directly stated yet are critical to comprehending the general sentiment of the text. This results in a more complex and thorough study.

Support for Multiple Languages: This method may be used to various languages, considering the linguistic variations and cultural settings that impact how feelings are implicitly expressed across languages.

Enhanced Natural Language Understanding (NLU): This method may be modified to function with many languages, considering the subtle linguistic variations and cultural backgrounds that impact the implicit expression of feelings in different languages.

Improved Accuracy in Sentiment Analysis: The accuracy of sentiment analysis may be significantly improved by considering implicit aspects. Traditional sentiment analysis approaches might miss or misinterpret sentiments associated with implicitly mentioned aspects, leading to less accurate results.

1.7 Terminologies

Sentiment analysis for hotel reviews - Sentiment analysis for hotel reviews is a specialized application of sentiment analysis aimed at understanding and categorizing the emotions and opinions expressed by guests in their reviews of hotel experiences.

This analysis provides valuable insights for both consumers and hotel management, influencing decision-making, service improvements, and strategic marketing.

Sentiment classification for hotel reviews - Sentiment classification for hotel reviews involves categorizing text data (reviews) into sentiment categories, typically positive, negative, or neutral. This process helps hotels and potential guests quickly understand the overall sentiment of feedback without reading through each review.

Hotel aspects - In the context of sentiment analysis for hotel reviews, focusing on specific hotel aspects can provide more detailed insights into guest experiences and satisfaction levels. Analyzing sentiment by aspect allows hotels to pinpoint areas of strength and opportunities for improvement. In this study, hotel aspects are staff service, cleanliness, value price, and convenience of location.

Implicit aspects in hotel reviews - Implicit aspects in hotel reviews refer to underlying themes or features that are not directly mentioned by the reviewer but can be inferred from the overall sentiment or context of the review. Unlike explicit aspects where the reviewer clearly mentions specific features (like “cleanliness” or “service”), implicit aspects require interpretation and deeper analysis to uncover. Identifying and analyzing these implicit aspects is crucial for a comprehensive understanding of guest sentiment and satisfaction.

Aspect-based sentiment analysis for hotel reviews - In the context of hotel reviews, ABSA is particularly useful as it allows for the extraction and analysis of sentiments related to specific aspects of the hotel experience, such as cleanliness, service quality, location, room comfort, amenities, and price. By identifying sentiments about these aspects, hotels can gain a nuanced understanding of guest satisfaction and areas for improvement.

CHAPTER II

LITERATURE REVIEWS

2.1 Customer Reviews

2.1.1 Definition

Customer reviews [5] are the opinions and comments of individuals who have purchased and utilized a product or service. Typically, these evaluations are written by consumers and can be found on various platforms, such as e-commerce websites, social media, review websites, and others. An example of customer reviews for a restaurant can be shown as follows.

“I had a wonderful dining experience at XYZ Restaurant last night. The food was exquisite, and the service was top-notch. I highly recommend trying their signature dish, it's a flavor explosion!”

An example of customer reviews for a product can be shown as follows.

“I purchased the XYZ Smartphone, and I'm thrilled with it. The camera quality is amazing, and the battery life exceeded my expectations. It's definitely worth the price.”

2.1.2 Benefits of Customer Review

The benefits of customer evaluations may include [5][13]:

Information and Insight: Customer reviews enlighten future purchasers about a product or service from the perspective of people who have already used it. This information may contain specifics regarding the product's features, quality, performance, and durability, among other things.

Trust and Credibility: Positive evaluations can increase a product or company's credibility and trustworthiness. When prospective consumers observe the positive experiences of others, they may be more inclined to make a purchase.

Decision-Making: Reviews play a crucial role in enabling customers to make well-informed decisions. Individuals have the ability to evaluate the advantages and disadvantages of a product by considering the experiences of others, enabling them to ascertain its compatibility with their specific requirements and expectations.

Feedback for Improvement: Customer reviews may offer organizations with useful information. This input may be used by businesses to discover areas for improvement, address client issues, and improve their goods or services.

Marketing and Promotion: Positive feedback may be utilized in marketing and promotion. Businesses frequently highlight positive evaluations in promotional materials in order to attract more consumers.

Community and Interaction: Customer testimonials can cultivate a sense of community among consumers. People frequently read and respond to reviews in order to share their own experiences, pose queries, or offer advice.

Rankings and Search Engine Optimization: Many online platforms use user evaluations as part of their ranking algorithms. Products or services with higher ratings and more favorable evaluations may be prioritized in search engine results.

2.1.3 Hotel Customer Reviews

Hotel customer reviews [14] refers to the feedback and opinions expressed by those who have remained at a specific hotel. Customers can share their experiences, impressions, and evaluations of a hotel's amenities, services, sanitation, staff, location, and overall quality on a variety of online platforms, travel websites, and booking platforms. An example of hotel customer review can be shown as follows.

“My stay at the ABC Hotel was fantastic. The room was spacious and clean, and the staff was incredibly helpful and friendly. The hotel's location was convenient for exploring the city.”

Hotel customer reviews are essential for both hospitality businesses and potential guests. Here is a more in-depth examination of their significance and potential impact:

1. Influence on Booking Decisions: When selecting a hotel, many travelers rely on customer feedback. Negative reviews can deter potential visitors, while positive reviews can increase bookings. Reviews provide travelers with vital information for making informed decisions.

2. Trust and Credibility: Authentic reviews from former customers increase a hotel's confidence and trustworthiness. Potential visitors are more inclined to trust fellow travelers' comments and experiences than promotional materials from the hotel itself.

3. Feedback for Improvement: Reviews provide hotels with valuable information regarding their services, amenities, and overall guest satisfaction. This feedback can assist hotels in identifying areas for enhancement and implementing the necessary adjustments to increase client satisfaction.

4. Guest Engagement: Responding to both positive and negative evaluations demonstrate a hotel's dedication to ensuring client satisfaction. This type of interaction with visitors can result in enhanced relationships and brand loyalty.

5. Quality Control: Reviews function as a quality control mechanism. Consistently positive evaluations can be used as evidence of a hotel's dedication to providing a positive visitor experience.

6. Insights into Guest Preferences: By analyzing reviews, hotels can gain insight into guest preferences, such as which amenities are most essential to them, which can inform future business and marketing decisions.

7. Competitive Advantage: Positive reviews can provide a hotel with a competitive edge over its competitors. A hotel's ratings and evaluations can distinguish it from others in the same location or price range.

8. Marketing Tool: In marketing materials and on their websites, hotels can demonstrate the quality of their services by highlighting positive consumer feedback. This can be particularly effective in attracting new guests.

9. Risk Management: Negative reviews can alert hotels to potential issues that need to be addressed, allowing them to better manage risks and prevent the emergence of larger problems.

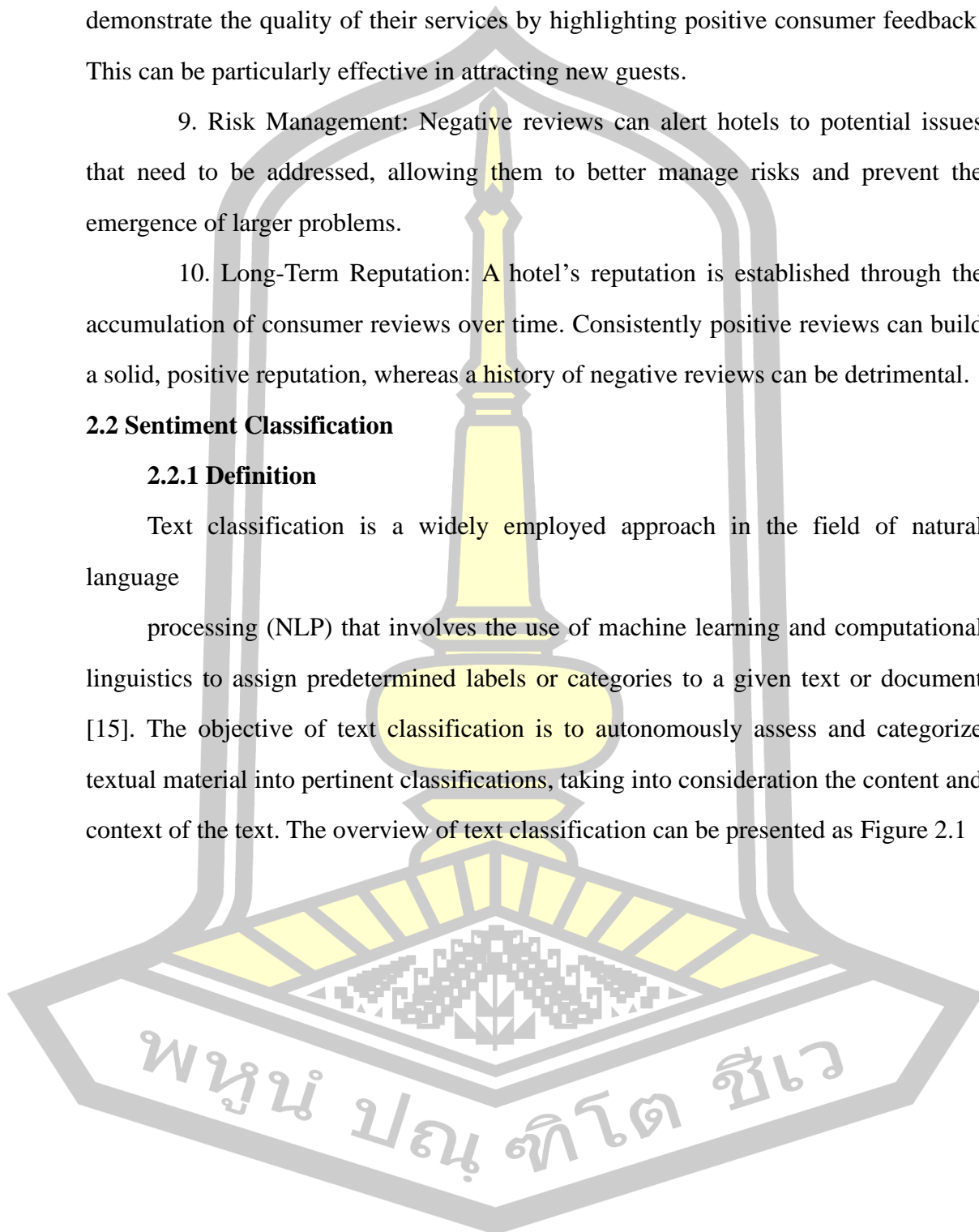
10. Long-Term Reputation: A hotel's reputation is established through the accumulation of consumer reviews over time. Consistently positive reviews can build a solid, positive reputation, whereas a history of negative reviews can be detrimental.

2.2 Sentiment Classification

2.2.1 Definition

Text classification is a widely employed approach in the field of natural language

processing (NLP) that involves the use of machine learning and computational linguistics to assign predetermined labels or categories to a given text or document [15]. The objective of text classification is to autonomously assess and categorize textual material into pertinent classifications, taking into consideration the content and context of the text. The overview of text classification can be presented as Figure 2.1



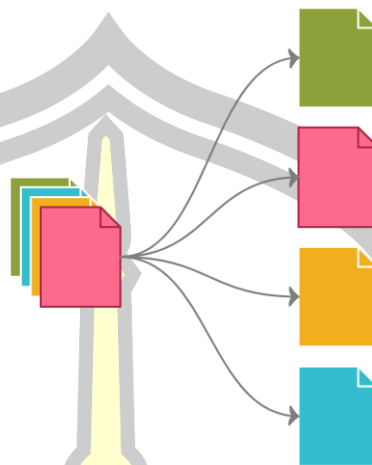


Figure 2.1 The overview of text classification

From: <https://dev.to/aashish/text-analytics-text-classification-2b72>

Sentiment classification is an is a specific use case of text classification. In sentiment classification, the sentiment is often classified as good, negative, or neutral. As a result, sentiment classification can employ the text classification approach. Today, sentiment classification is a critical method in a wide range of applications, including social media monitoring, customer feedback analysis, and market research.

2.2.2 Types of Sentiment Classification

Text classification can be classified into three distinct approaches to categorizing and analyzing text data. [16]

Document-based Text Classification[17]-Document-based text classification involves categorizing documents according to their content into predetermined groupings or categories. The method is essential for many applications, including as sentiment analysis, topic labeling, spam detection, and more. The goal is to automatically assign labels or categories to entire documents, making it easier to manage, sort, and utilize large volumes of text data.

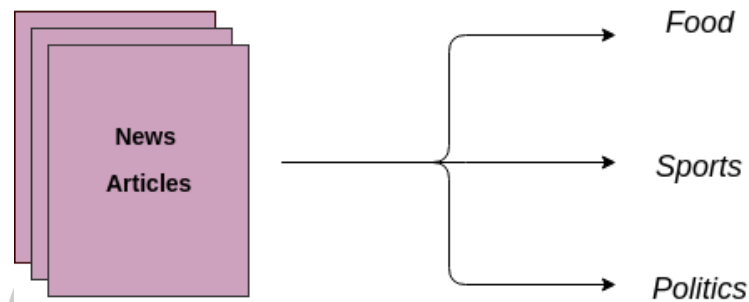


Figure 2.2 The overview of document-based text classification

From: <https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/>

Sentence-based Text Classification[17] - Sentence-based text classification focuses on categorizing individual sentences into predefined categories or labels based on their content. Unlike document-based classification, which deals with larger text units, sentence-based classification handles finer-grained text elements, making it suitable for applications like sentiment analysis at the sentence level, intent recognition in user queries, or identifying specific types of statements (e.g., questions, commands) within larger texts.

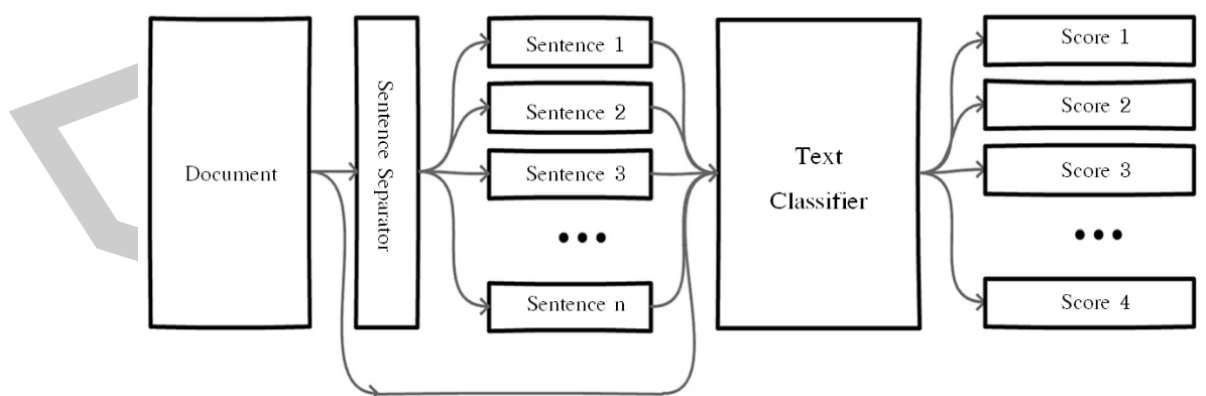


Figure 2.3 The overview of Sentence-based text classification

Aspect-based Text Classification [18] - Aspect-based text classification (ABTC) is a more granular approach within the realm of text analysis, aiming to categorize text not just by overall sentiment or topic, but with respect to specific aspects or attributes mentioned within the text. This technique is particularly useful in sentiment analysis, where the goal is to understand the sentiment towards specific features or aspects of a product, service, or topic. For example, in analyzing restaurant reviews, ABTC can distinguish sentiments expressed about the food quality, service, ambiance, and price, rather than giving a single sentiment score for the entire review.

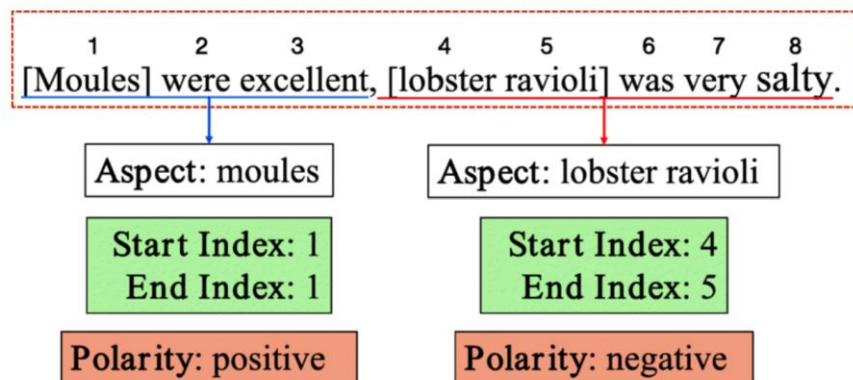


Figure 2.4 The overview of aspect-based text classification

From: https://www.researchgate.net/publication/373015779_RGB-T_salient_object_detection_via_excavating_and_enhancing_CNN_features

2.3 Aspect-based Text Classification (ABTC)

2.3.1 Challenges and Considerations

ABTC offers a nuanced understanding of texts by dissecting them into their constituent sentiments or opinions about specific aspects. This approach is invaluable for businesses and researchers looking to gain detailed insights into customer feedback or public opinion on various topics. However, there are many significant challenges for ABTC. Here, these are some challenges for ABTC [19][20][21][22].

1. Aspect Ambiguity [21]: Determining the aspect a segment of text refers to can be challenging, especially when the text mentions multiple aspects or uses vague

language. Consider following example sentence, “She saw him with the telescope.”. From this aspect, “telescope” is interpreted as an optical instrument used for viewing distant objects. In this interpretation, “She” used the telescope to see “him,” implying that “he” was far away and needed assistance to be seen clearly. In addition, “telescope” can be interpreted as an event related to astronomy, such as observing stars or planets through a telescope. In this interpretation, “She” is the one being observed by “him” using the telescope.

2. Data Sparsity [22]: For some aspects, there may be relatively few mentions in the available data, making it difficult to train models effectively for those aspects. Suppose we have collected 1000 restaurant reviews for training our sentiment analysis model. Within these reviews, we have the following distribution of mentions for each aspect: (1) Food Quality: 800 reviews, (2) Service: 700 reviews, (3) Ambiance: 50 reviews, and (4) Price: 600 reviews. In these examples, data sparsity is observed for the “Ambiance” aspect, as there are only 50 reviews that mention it compared to a more abundant number of reviews for other aspects. As a result, the model may not have enough training examples to effectively learn the patterns and nuances associated with ambiance-related sentiments. This can lead to poorer performance in classifying sentiment for the ambiance aspect compared to other aspects.

3. Contextual Dependency [22]: The sentiment or category of an aspect often depends heavily on the context, requiring models to have a strong ability to understand nuanced language use. An example can be presented as follows. Consider a review of a restaurant, “Although the food was delicious, the service was slow.” In this review, there are two aspects mentioned: food quality and service. The sentiment towards each aspect can be different and is dependent on the surrounding context.

- Food Quality Aspect: The phrase “*the food was delicious*” indicates a positive sentiment towards food quality.
- Service Aspect: The phrase “*the service was slow*” suggests a negative sentiment towards the service aspect.

However, the sentiment towards each aspect is heavily influenced by the surrounding context. If we were to remove the contrasting aspect, the sentiment towards the remaining aspect might change. For instance:

- If we only consider “*the food was delicious*” without the mention of service, it might be interpreted more positively as it's not compared with a negative aspect.
- Similarly, if we only consider “*the service was slow*” without the mention of food quality, it might seem even more negative as there's no contrasting positive aspect.

This example highlights how the sentiment towards different aspects within a text can be contextually dependent. In aspect-based text classification, models need to understand and consider the surrounding context to accurately classify sentiments for each aspect. This requires capturing dependencies between aspects and their surrounding context to make more nuanced predictions.

2.3.2 Generic Framework for Aspect-based Text Classification

The essential components of ABTC may be seen in Figure 5. A brief explanation may be provided for each processing step [23][24][25].

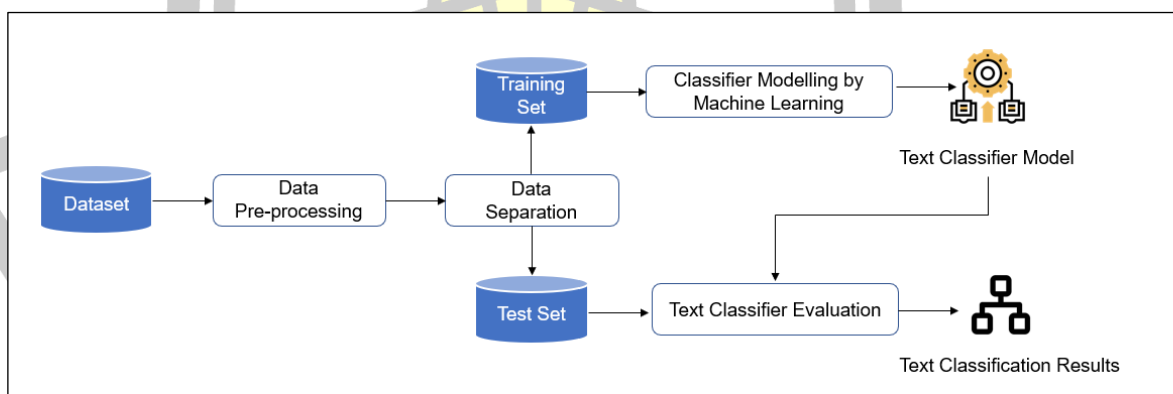


Figure 2.5 generic framework of aspect-based text classification

1. Aspect Identification – This process is a critical initial step in ABTC, where the goal is to recognize and categorize specific aspects or attributes mentioned within a text. This process enables a more nuanced analysis, such as determining sentiments towards particular features of a product or service rather than assessing the text's overall sentiment.

(1) Extraction of Aspects [23]: The first step involves identifying the different aspects mentioned in the texts. This can be achieved through techniques such as keyword extraction, topic modeling (e.g., Latent Dirichlet Allocation), or using predefined lists of aspects relevant to the domain. Existing approaches can be presented as follows.

- Manual Identification [24]: It involves creating a list of aspects based on domain knowledge. This approach is straightforward but can be time-consuming and may not capture all relevant aspects, especially in diverse datasets.

- Keyword or Rule-based Extraction [25][26]: It is to utilize predefined keywords and linguistic rules to identify aspects. While this method can effectively capture known aspects, it may miss nuanced or contextually defined aspects.

- Supervised Learning [27]: Machine learning models are trained on labeled datasets where aspects are annotated. Techniques like Support Vector Machines (SVM), Random Forests, or more advanced neural networks can be used. The limitation here is the need for a substantial amount of labeled data.

- Topic Modeling [28]: Unsupervised techniques like Latent Dirichlet Allocation (LDA) can discover topics within texts that correspond to aspects. These methods are useful for exploring large datasets without predefined aspect categories but may require interpretation to map topics to specific aspects.

- Dependency Parsing [29]: It is to analyzes the grammatical structure of sentences to find relationships between words, helping to identify aspects based on their context and how they are discussed within the text.

(2) Aspect Categorization [30]: After identifying potential aspects, the next step is to categorize them into broader aspect groups relevant to the analysis. For example, in restaurant reviews, individual mentions like “waiters,” “service time,” and “attentiveness” might all be categorized under a broader “Service” aspect.

Challenges in aspect identification can be (1) ambiguity and polysemy, (2) data sparsity, and (3) dynamic aspects. Ambiguity and polysemy are words or phrases that may have multiple meanings or may refer to different aspects depending on the context, making aspect identification challenging. Data sparsity is that certain aspects may be mentioned infrequently, making it difficult to identify and categorize them accurately without sufficient examples. Dynamic aspects are new aspects that can emerge over time, especially in rapidly evolving fields, requiring continuous updating of the models and aspect categories.

2. Pre-processing - Similar to other text classification tasks, preprocessing steps like text cleaning, tokenization, normalization, and potentially stopword removal are applied to prepare the text for analysis.

3. Feature Extraction - Two possible techniques that can be applied for feature extraction for ABTC can be explained as follows.

(1) Contextual Embeddings [31]: Given the importance of context in determining the sentiment towards a specific aspect, contextual word embeddings from models like BERT, ELMO, or GPT are highly effective. They capture the nuanced semantic meaning of text segments related to each aspect.

(2) Syntactic Parsing [32]: Identifying relationships between words can also help in associating sentiments with specific aspects, for which dependency parsing might be used.

4. Model Selection and Training – These are possible approaches that can be applied for ABTC.

(1) Machine Learning Models [33]: Traditional models can be applied to ABTC, especially when used with feature engineering to highlight aspect-related information.

(2) Deep Learning Models [34][35]: Neural networks, particularly those that can handle sequential and contextual information (e.g., LSTM, GRU, Transformers), are well-suited for ABTC. The ability of these models to understand context and complex relationships within text makes them powerful for distinguishing sentiments towards different aspects.

(3) Transformer Learning Models [36]: Leveraging pre-trained models like BERT for ABTC by fine-tuning them on domain-specific datasets has become a common and effective approach.

5. Evaluation [37] - Evaluation metrics such as accuracy, precision, recall, and F1-score are used, often separately for each aspect to understand the model's performance in identifying and classifying sentiments or categories for each aspect specifically.

2.3.3 Pre-processing for Aspect-based Text Classification

Pre-processing for aspect-based text classification entails a series of activities aimed at cleansing and organizing the text data prior to its utilization in training a classification model. The following are the typical pre-processing steps:

- Tokenization [38] - The objective of the task is to divide the text into separate words or tokens. The process is crucial for subsequent analysis and extraction of features.
- Lowercasing [39] - The objective of this task is to transform all text into lowercase in order to maintain uniformity in word representations. This facilitates the treatment of words with varying cases (e.g., “Food” vs. “food”) as identical tokens.
- Removing Noise [40] - The objective of this task is to remove unnecessary characters, such as special symbols, punctuation marks, HTML tags, or non-alphanumeric characters, that are not relevant to the classification task.

- Stopword Removal [41] - The objective of this task is to eliminate often used stopwords (such as “the”, “is”, and “and”) that frequently appear in language but lack extensive semantic significance. Eliminating stopwords decreases the number of dimensions in the data and enhances computational performance.

- Stemming or Lemmatization [42] - The objective of this task is to standardize variations of the same word by reducing them to their base or root form. Stemming, such as the Porter Stemmer, removes prefixes and suffixes from words, whereas lemmatization converts words to their base form found in the dictionary. This stage helps in reducing sparsity and enhancing the comprehensiveness of the lexicon.

- Handling Negations [43] - The objective of this task is to identify negations (such as “not” and “never”) and alter the subsequent sentences to indicate the opposite attitude. For instance, the sentence "not good" can be converted to “not_good” in order to maintain the negation impact during sentiment analysis.

- Handling Aspect Terms [44] - The objective of this task is to recognize and standardize aspect phrases (such as “food quality” and “service”) present in the given text. Aspect terms refer to the precise elements or characteristics that are being assessed in aspect-based sentiment analysis.

- Feature Representation [45] - This task is to represent the pre-processed text data using suitable feature representations, such as Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), or word embeddings (e.g., Word2Vec, GloVe). These representations capture the semantic information of the text and enable machine learning models to learn patterns effectively.

2.3.4 Classifier Modeling for Aspect-based Text Classification

The purpose of this stage is to select a suitable classification algorithm or model for aspect-based text classification. Commonly employed models include:

2.3.4.1 Machine Learning

Several machine learning algorithms are applicable for this purpose. Below are some commonly used algorithms:

▪ *Multinomial Naive Bayes (MNB)* [46][47] - MNB is a probabilistic classifier based on Bayes' theorem with strong independence assumptions between features. It is particularly well-suited for text classification tasks, including sentiment analysis and document categorization. This algorithm assumes that features (words or tokens in text classification) are generated from a multinomial distribution. It calculates the conditional probability of a document belonging to a particular class (e.g., positive sentiment) given its features (words). To classify a new document, MNB applies Bayes' theorem to calculate the posterior probability of each class given the document's features. The class with the highest posterior probability is selected as the predicted class for the document. This algorithm is simple, fast, and robust to irrelevant features. It performs well in practice, especially for text classification tasks with sparse, high-dimensional feature spaces. However, it may not capture complex relationships between features. The independence assumption may not hold true for all datasets, leading to suboptimal performance in some cases.

This algorithm first determines the percentage of documents in each class, denoted as $P(c)$, and then determines the likelihood of each word for a certain class, denoted as $P'(w_i | c)$, to create the classifier model. These formulas can be written as Equation (2.1).

$$P(c) = \frac{N_{class}}{N} \quad (2.1)$$

where N is the total number of reviews in the training set, and N_{class} is the total number of reviews in each class, and

$$P'(w_i | c) = \frac{count(w_i, c) + \alpha}{count(c) + |V| + 1} \quad (2.2)$$

where $count(w, c)$ shows how many times the term w is found in the class c . In the meantime, $count(c)$ denotes the total number of training set classes, and $|V|$ denotes the total number of distinct words inside the training set. Since some words

have zero counts, Laplace smoothing is performed with a low value of $\alpha = 0.001$. Finally, the Bayes' rule is used to calculate an estimate of $P'(c | d)$ for the test documents. The prediction formula can be written as Equation (2.3).

$$P'(c | d) = \arg \max P(c) \prod_{i=1}^n P'(w_i | C_j) \quad (2.3)$$

Logistic Regression (LR) [48] - LR is a linear model used for binary classification tasks. It estimates the probability that a given input belongs to a particular class using a logistic function. LR models the relationship between the input features (text features representing aspects or features mentioned in the text) and the probability of each sentiment class (e.g., positive, negative) using a linear function. It uses the sigmoid (logistic) activation function to transform the linear combination of input features into a probability score between 0 and 1. This probability score represents the likelihood of the input belonging to a particular sentiment class. LR can be applied to both binary and multi-class classification tasks. For binary classification, a single logistic regression model is trained to distinguish between two sentiment classes (e.g., positive vs. negative). For multi-class classification, multiple logistic regression models are trained by using techniques like one-vs-rest or multinomial logistic regression, each representing a different sentiment class, and the class with the highest probability is predicted. To prevent overfitting and improve generalization performance, LR can incorporate regularization techniques such as L1 regularization (Lasso) or L2 regularization (Ridge). Regularization penalizes large weights in the model, encouraging simpler models with better generalization properties.

The LR algorithm can be utilized to solve classification problems by establishing thresholds for the probability predicted for each class [48]. Although this algorithm is commonly used for binary classification, it can easily adapt to multiple classes. In LR, given N bug reports x_i , $i = 1, \dots, N$, the m features of the input bug report $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$, are linearly integrated using coefficient β_0 and $\beta =$

$(\beta_0, \dots, \beta_m)$ to predict the classification outcome y_i . Specifically, given an input bug report x_i , the probability that $y_i = 1$ is indicated by $P(x_i)$ and is modelled with the conventional logistic regression model as Equation (2.4).

$$P(x_i) = \frac{e^{(\beta_0 + \beta^* x_i)}}{1 + e^{\beta_0 + \beta^* x_i}} \quad (2.4)$$

The LR classifiers employ a sigmoid function to process the weighted combination of input features. By using the sigmoid function, any real value can be converted to a number between 0 and 1.

Support Vector Machines (SVM) [49] - SVM is a powerful supervised learning algorithm that can be used for both binary and multi-class classification tasks. SVM aims to find the hyperplane that maximizes the margin between classes. SVM with linear or kernelized versions can be applied to aspect-based text classification, effectively separating sentiment categories for different aspects. SVMs can perform both linear and non-linear classification. In aspect-based text classification, SVMs learn a decision boundary that separates text samples into different sentiment categories based on the specific aspects or features mentioned. This algorithm aims to find the hyperplane that maximizes the margin between different classes in the feature space. The margin is the distance between the hyperplane and the nearest data points (support vectors) from each class. Maximizing the margin helps SVM generalize well to unseen data and improves robustness. SVMs can handle non-linear decision boundaries by using kernel functions that map the input features into a higher-dimensional space where the classes become linearly separable. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels. SVMs incorporate regularization to control model complexity and prevent overfitting. Regularization parameters, such as the C parameter, control the trade-off between maximizing the margin and minimizing classification errors on the training data.

Let x_1, x_2, \dots, x_l be training examples belonging to one class C , where C is a compact subset of \mathbb{R}^N . The SVM classifier can be built using Equation (2.5).

Support Vector Machines (SVM) [49] - SVM is a powerful supervised learning algorithm that can be used for both binary and multi-class classification tasks. SVM aims to find the hyperplane that maximizes the margin between classes. SVM with linear or kernelized versions can be applied to aspect-based text classification, effectively separating sentiment categories for different aspects. SVMs can perform both linear and non-linear classification. In aspect-based text classification, SVMs learn a decision boundary that separates text samples into different sentiment categories based on the specific aspects or features mentioned. This algorithm aims to find the hyperplane that maximizes the margin between different classes in the feature space. The margin is the distance between the hyperplane and the nearest data points (support vectors) from each class. Maximizing the margin helps SVM generalize well to unseen data and improves robustness. SVMs can handle non-linear decision boundaries by using kernel functions that map the input features into a higher-dimensional space where the classes become linearly separable. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels. SVMs incorporate regularization to control model complexity and prevent overfitting. Regularization parameters, such as the C parameter, control the trade-off between maximizing the margin and minimizing classification errors on the training data.

Let x_1, x_2, \dots, x_l be training examples belonging to one class C , where C is a compact subset of \mathbb{R}^N . The SVM classifier can be built using Equation (2.5).

Random Forest (RF) [50] - Random Forest is an ensemble learning method that builds multiple decision trees during training and combines their predictions through voting or averaging. Random Forest can handle high-dimensional feature spaces and capture complex interactions between features, making it suitable for aspect-based sentiment analysis tasks. This algorithm employs a technique called bootstrap aggregation (bagging), where each decision tree is trained on a bootstrapped

sample of the original training data. This sampling with replacement creates diverse subsets of the data, reducing overfitting and improving. In addition to sampling data points, RF also introduces randomness in feature selection by considering only a subset of features at each split in the decision tree. This randomness ensures that each tree in the forest learns different aspects or features of the input text data. Each decision tree in the Random Forest is constructed by recursively partitioning the feature space based on the selected features and splitting criteria (e.g., Gini impurity, information gain). The trees grow deep enough to capture complex relationships between input features and class labels. During inference, the predictions of individual decision trees are aggregated to obtain the final prediction. For classification tasks, the class with the most votes among the trees is chosen as the predicted class. For regression tasks, the average of the individual tree predictions is taken. RF promotes ensemble diversity by training each decision tree independently and introducing randomness in both data and feature selection. This diversity helps prevent overfitting and improves the robustness of the model to noisy or irrelevant features. RF are relatively efficient and scalable, capable of handling large datasets with high-dimensional feature spaces. However, the training and inference times may increase with the number of trees in the forest, especially for very large forests.

Gradient Boosting Machines (GBM) [51] - GBM is another ensemble learning technique that builds a sequence of weak learners (typically decision trees) and combines them to improve predictive performance. GBM sequentially minimizes a loss function by adding new models that correct errors made by existing models. GBM variants like XGBoost [52], LightGBM [52], and CatBoost [52] have been successfully applied to aspect-based text classification tasks due to their efficiency and effectiveness. GBM commonly uses decision trees as weak learners. Each decision tree is trained to predict the residual errors (i.e., the difference between the predicted and actual values) of the previous models. By combining multiple decision trees, GBM creates a strong learner that captures complex relationships in the data.

This algorithm optimizes model parameters (e.g., tree structure, weights) using gradient descent or a similar optimization algorithm. During each iteration, GBM minimizes a loss function (e.g., mean squared error for regression, cross-entropy loss for classification) by moving in the direction of steepest descent in the model's parameter space.

2.3.4.2 Deep Learning

Deep learning algorithms have demonstrated remarkable performance in various natural language processing tasks, including aspect-based text classification [53][54][55]. Here are some commonly used deep learning algorithms for aspect-based text classification:

1. Convolutional Neural Networks (CNNs) [55][56] - CNNs are effective models for aspect-based text classification, leveraging their ability to capture local patterns and hierarchical representations from input text data. By extracting relevant features and modeling dependencies between aspects and their context, CNNs can accurately classify sentiments or opinions associated with specific aspects mentioned in the text. Below is a brief overview of CNN's application in aspect-based text classification:

- 1) Input Representation - In text classification, the input data consists of sequences of words or characters. Each word or character is typically represented as a vector through techniques like word embeddings (e.g., Word2Vec, GloVe) or character embeddings. These embeddings capture the semantic meaning or character-level information of the input text.

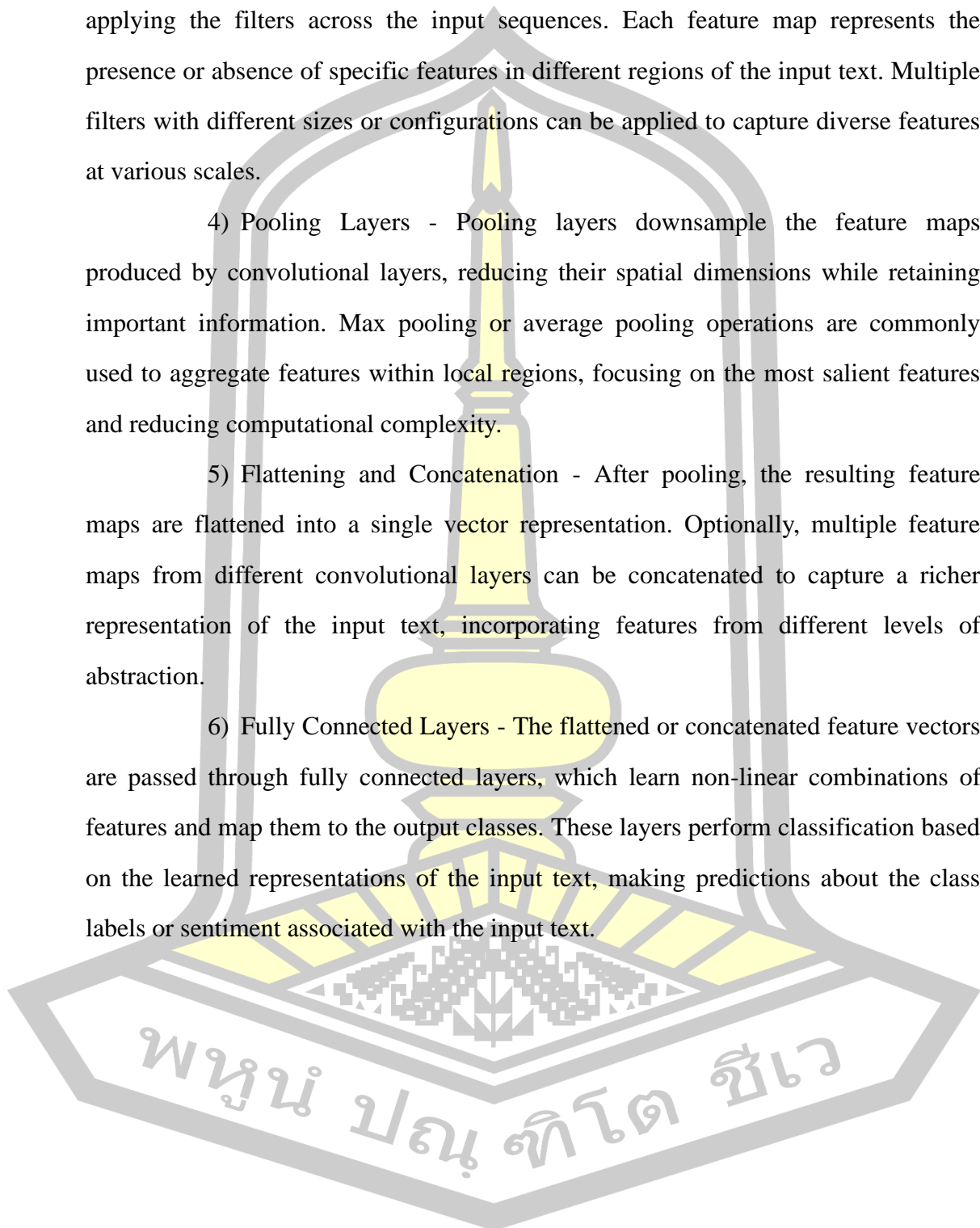
- 2) Convolutional Layers - Similar to image processing, CNNs use convolutional filters to extract local patterns or features from the input text. The filters slide over the input sequences, capturing important features such as n-grams or word sequences. The size of the filters determines the size of the local context captured by each filter.

3) Feature Maps - Convolutional filters generate feature maps by applying the filters across the input sequences. Each feature map represents the presence or absence of specific features in different regions of the input text. Multiple filters with different sizes or configurations can be applied to capture diverse features at various scales.

4) Pooling Layers - Pooling layers downsample the feature maps produced by convolutional layers, reducing their spatial dimensions while retaining important information. Max pooling or average pooling operations are commonly used to aggregate features within local regions, focusing on the most salient features and reducing computational complexity.

5) Flattening and Concatenation - After pooling, the resulting feature maps are flattened into a single vector representation. Optionally, multiple feature maps from different convolutional layers can be concatenated to capture a richer representation of the input text, incorporating features from different levels of abstraction.

6) Fully Connected Layers - The flattened or concatenated feature vectors are passed through fully connected layers, which learn non-linear combinations of features and map them to the output classes. These layers perform classification based on the learned representations of the input text, making predictions about the class labels or sentiment associated with the input text.



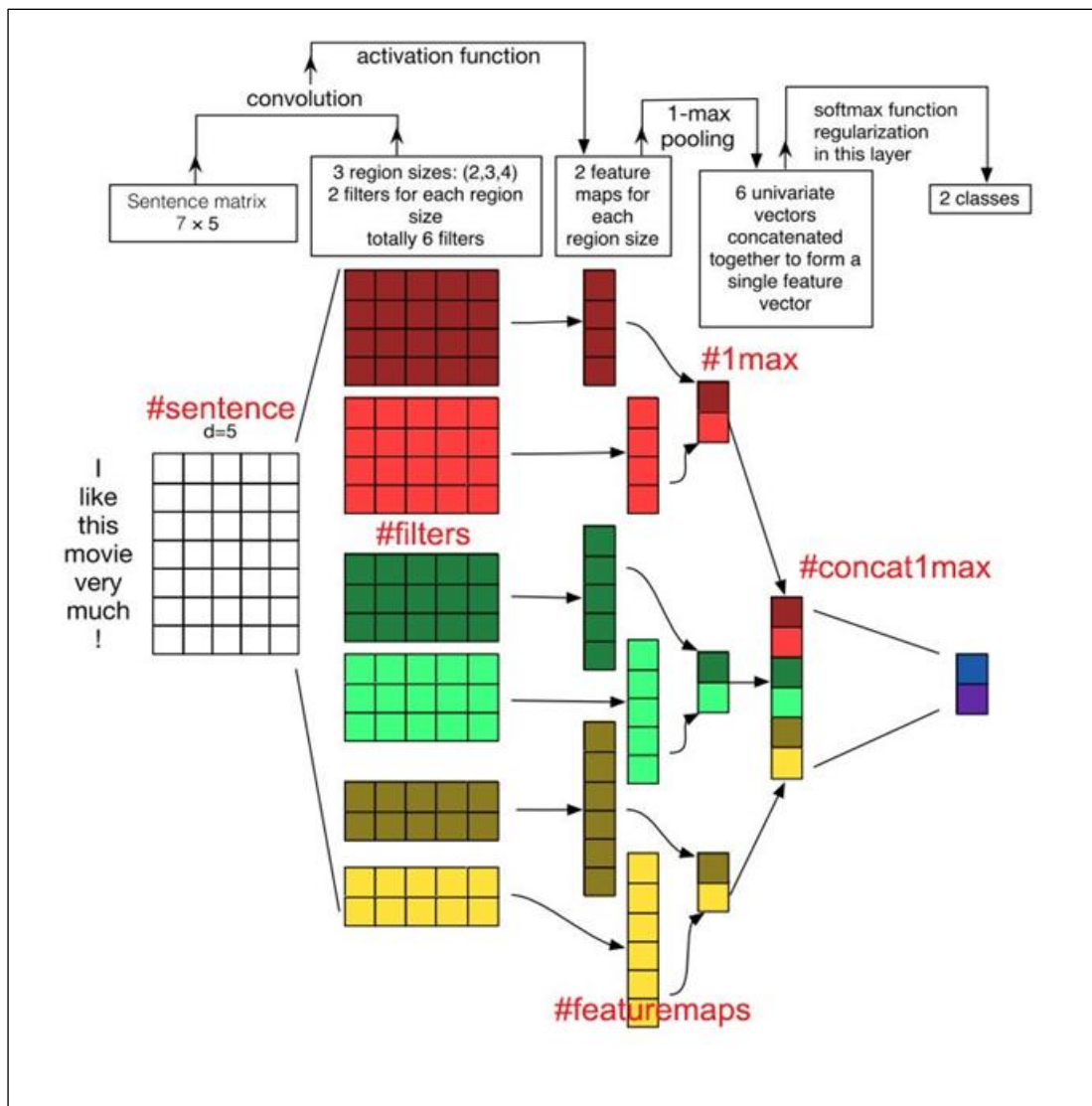


Figure 2.6 An overview of CNN structure for text classification

From: <https://towardsdatascience.com/understanding-how-convolutional-neural-network-cnn-perform-text-classification-with-word-d2ee64b9dd0b>

7) Activation Functions and Regularization - Activation functions like ReLU (Rectified Linear Unit) introduce non-linearity into the model, enabling it to learn complex relationships between features. Regularization techniques such as dropout may be applied to prevent overfitting by randomly dropping neurons during training.

8) Loss Function and Optimization - CNNs are trained using standard optimization techniques such as stochastic gradient descent (SGD) or adaptive optimization algorithms like Adam. The loss function used for training depends on the specific task, such as binary cross-entropy for binary classification or categorical cross-entropy for multi-class classification.

By leveraging local patterns and hierarchical representations in the input text, CNNs can effectively capture semantic information and learn discriminative features for text classification tasks. With appropriate architecture design and hyperparameter tuning, CNNs have demonstrated state-of-the-art performance in various text classification benchmarks.

2. Recurrent Neural Networks (RNNs) [54][57]- RNNs are a class of neural networks particularly well-suited for sequential data processing tasks, including text classification. Here's how RNNs can be used for text classification:

1) Sequential Modeling - RNNs are designed to process sequential data by maintaining a hidden state that captures the context of the input sequence up to the current time step. This sequential nature makes RNNs suitable for tasks where the order of input elements (e.g., words in a sentence) is important, such as text classification.

2) Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) - Variants of RNNs, such as LSTMs [58][59] and GRUs [60], address the vanishing gradient problem associated with traditional RNNs. LSTMs and GRUs incorporate gating mechanisms that control the flow of information through the network, enabling them to capture long-range dependencies and alleviate the issue of information loss over long sequences.

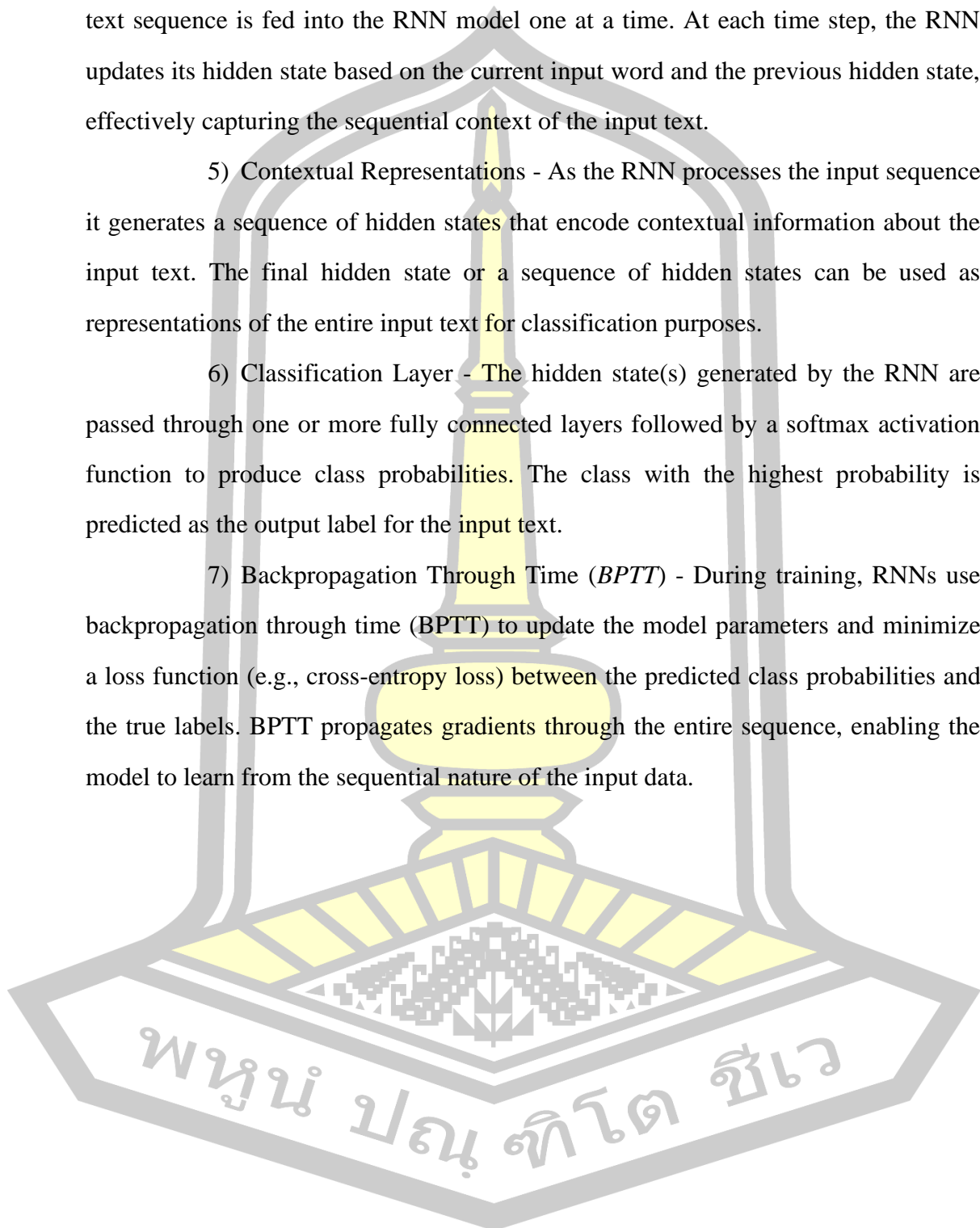
3) Word Embeddings - Input text data is typically represented using word embeddings, which map words to high-dimensional vector representations. These word embeddings capture semantic relationships between words and enable the model to learn meaningful representations of the input text.

4) Sequential Processing - In text classification, each word in the input text sequence is fed into the RNN model one at a time. At each time step, the RNN updates its hidden state based on the current input word and the previous hidden state, effectively capturing the sequential context of the input text.

5) Contextual Representations - As the RNN processes the input sequence, it generates a sequence of hidden states that encode contextual information about the input text. The final hidden state or a sequence of hidden states can be used as representations of the entire input text for classification purposes.

6) Classification Layer - The hidden state(s) generated by the RNN are passed through one or more fully connected layers followed by a softmax activation function to produce class probabilities. The class with the highest probability is predicted as the output label for the input text.

7) Backpropagation Through Time (*BPTT*) - During training, RNNs use backpropagation through time (*BPTT*) to update the model parameters and minimize a loss function (e.g., cross-entropy loss) between the predicted class probabilities and the true labels. *BPTT* propagates gradients through the entire sequence, enabling the model to learn from the sequential nature of the input data.



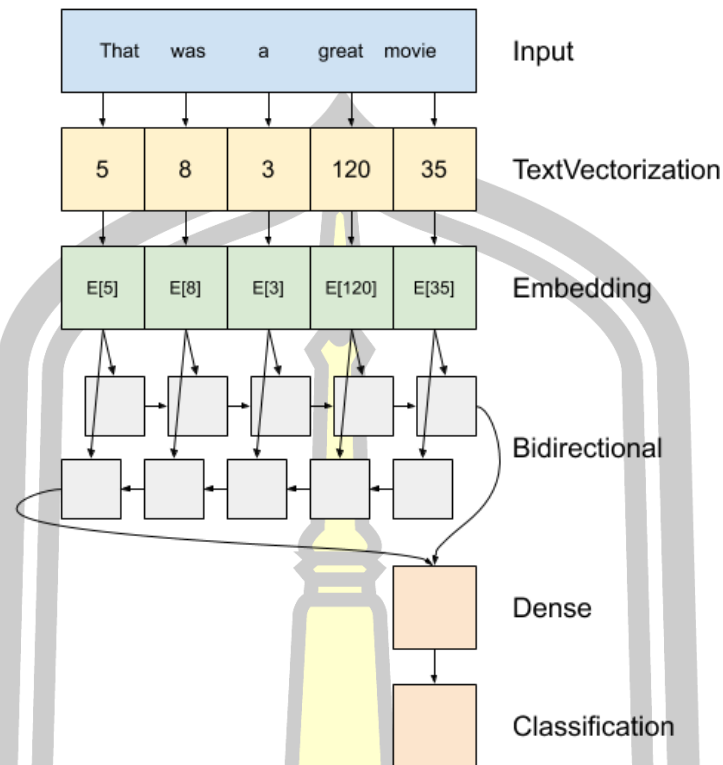


Figure 2.7 An overview of RNN structure for text classification

From: https://www.tensorflow.org/text/tutorials/text_classification_rnn?hl=th

Despite their effectiveness, RNNs may face challenges in handling long-range dependencies, vanishing/exploding gradients, and capturing complex patterns in text data. Techniques like gradient clipping, careful initialization, and architectural modifications (e.g., attention mechanisms) can help address these challenges. In general, RNNs may be powerful models for text classification tasks, capable of capturing sequential dependencies and contextual information in input text data. Variants like LSTMs and GRUs have been particularly successful in addressing the limitations of traditional RNNs, making them widely used in various NLP applications, including aspect-based text classification.

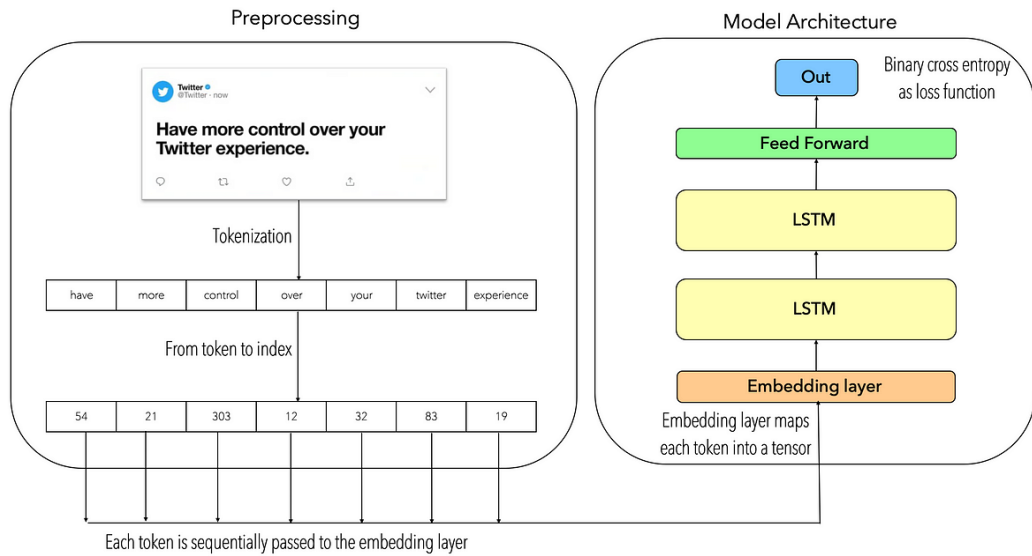


Figure 2.8 An overview of LSTM structure for text classification

From: <https://towardsdatascience.com/text-classification-with-pytorch-7111dae111a6>

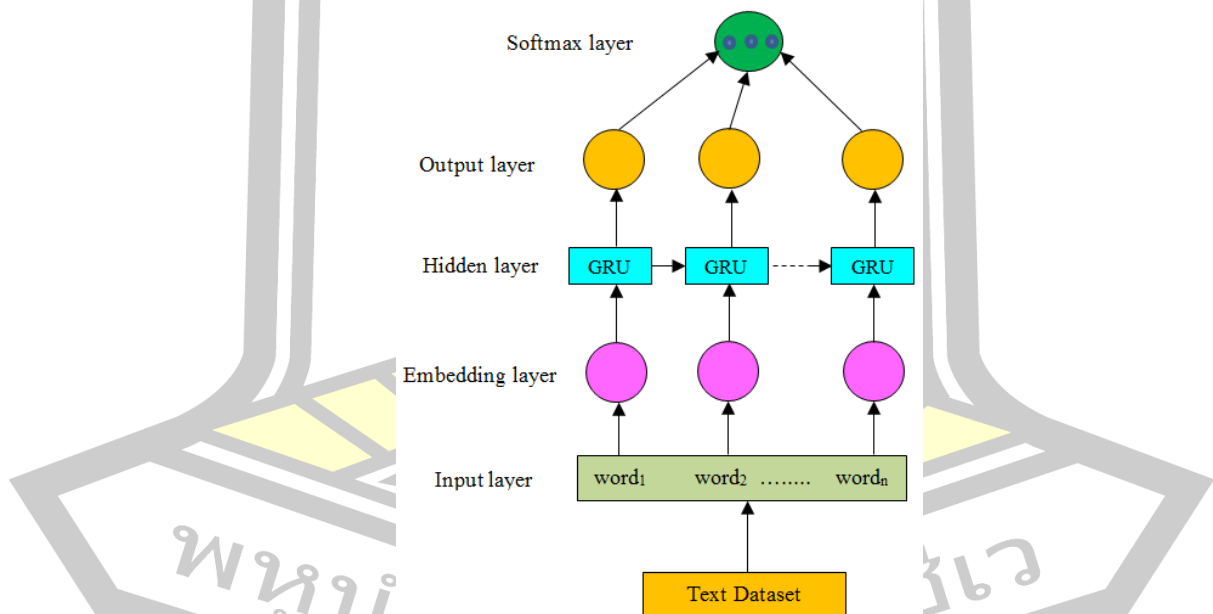


Figure 2.9 An overview of GRU structure for text classification

From: https://www.researchgate.net/figure/Architecture-of-GRU-model-for-text-classification_fig2_372670328

2.3.4.3 Ensemble Method

Ensemble methods [61][62] are techniques in machine learning that combine the predictions from multiple models to improve the overall performance. This approach is based on the idea that a group of weak learners can come together to form a strong learner. The rationale behind ensemble methods is that by combining different models, the strengths of each can be harnessed and their weaknesses mitigated, leading to more accurate and robust predictions. Ensemble methods can be broadly classified into three categories:

Bagging (Bootstrap Aggregating) [63][64]: Bagging involves training multiple models of the same type on different subsets of the training data and then averaging their predictions to obtain the final result. Each model is trained independently in parallel, which reduces variance and helps to avoid overfitting. A well-known example of bagging is the Random Forest algorithm, where multiple decision trees are trained on different subsets of the training data. Figure 2.10 An overview of bagging method presents an overview of Bagging method.

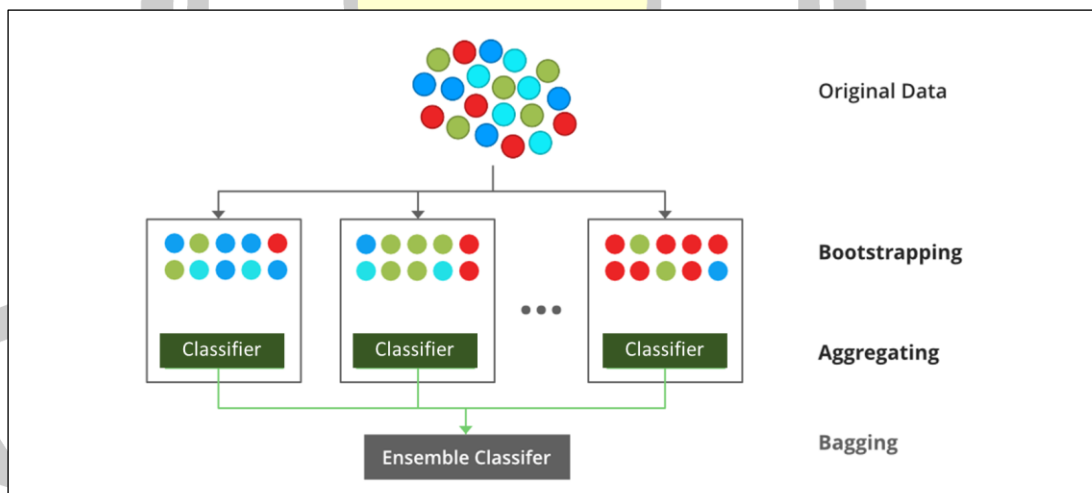


Figure 2.10 An overview of bagging method

From: <https://medium.com/@hemaanushatangellamudi/bootstrapped-aggregation-bagging-481f4812e3ea>

Boosting [65]: Boosting involves sequentially training multiple models, where each model learns from the errors of its predecessors to improve its accuracy. Unlike bagging, where models are trained independently, boosting focuses on training models that correct the mistakes made by previous ones, effectively reducing bias and variance. Examples of boosting algorithms include AdaBoost, Gradient Boosting, and XGBoost. Figure 2.11 presents an overview of Boosting method.

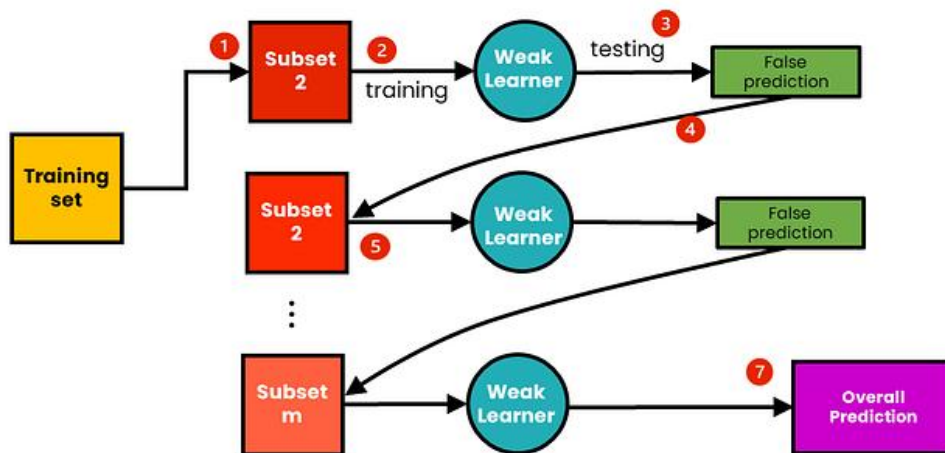


Figure 2.11 An overview of Boosting method

From: https://medium.com/@brijesh_soni/understanding-boosting-in-machine-learning-a-comprehensive-guide-bdeaa1167a6

Stacking (Stacked Generalization) [66] Stacking involves training a new model to combine the predictions of several base models. The base models are trained on the full training set, and then a new model is trained on the outputs of the base models as features. This method aims to leverage the strengths of each base model and can lead to significant improvements in prediction accuracy. Figure 2.11 presents an overview of Stacking method.

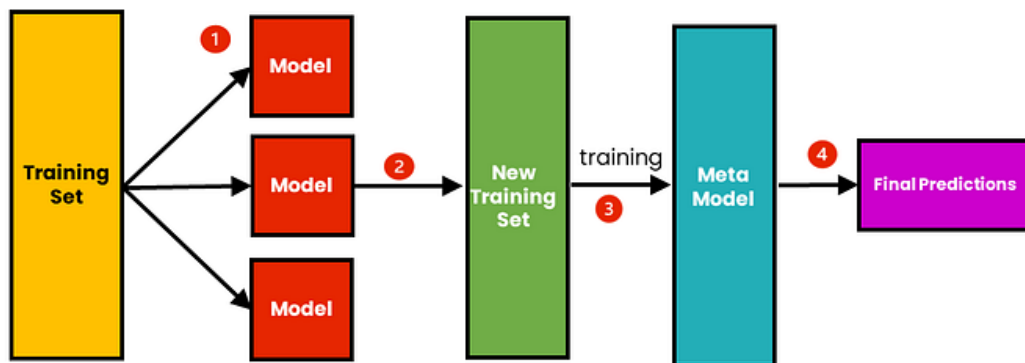


Figure 2.12 An overview of Stacking method

From: https://medium.com/@brijesh_soni/stacking-to-improve-model-performance-a-comprehensive-guide-on-ensemble-learning-in-python-9ed53c93ce28

Ensemble methods are widely used across various applications, such as classification, regression, and anomaly detection, due to their effectiveness in improving prediction accuracy and model robustness. They are particularly powerful in competitions and real-world problems where even a small increase in accuracy can be significant.

2.3.4.4 Transformer Learning

Transformer models [67][68][69] have revolutionized natural language processing (NLP) tasks, including text classification. Transformer learning can be utilized for text classification in the following manner:

1) Transformer Architecture - The transformer architecture, introduced in the paper “Attention is All You Need,” relies on self-attention mechanisms to capture long-range dependencies in input sequences. It consists of an encoder-decoder architecture, where the encoder is primarily used for text classification tasks.

2) Self-Attention Mechanism - Self-attention allows the model to weigh the importance of each word in the input sequence based on its relevance to other words. This mechanism enables transformers to capture contextual information

effectively, making them well-suited for text classification tasks where understanding the relationship between words is crucial.

3) Positional Encoding - Since transformers do not inherently understand the sequential order of input tokens, positional encodings are added to the input embeddings to provide information about the position of each token in the input sequence. This ensures that the model can differentiate between tokens based on their positions.

4) Input Representation - Input text data is typically tokenized into subword or word-level tokens, which are then converted into dense vector representations using pre-trained embedding models like Word2Vec, GloVe, or subword embeddings such as Byte Pair Encoding (BPE) or WordPiece embeddings.

5) Transformer Encoder - In text classification tasks, only the encoder part of the transformer model is typically used. The input embeddings are passed through multiple layers of self-attention mechanisms and feed-forward neural networks. The output of the encoder can be used as a representation of the input text for classification.

6) Pooling and Classification - The output of the transformer encoder can be aggregated using pooling operations such as mean pooling or max pooling to obtain a fixed-size representation of the input text. This representation is then passed through one or more fully connected layers followed by a softmax activation function for classification.

7) Pre-training and Fine-tuning - Transformer models are often pre-trained on large-scale text corpora using unsupervised learning objectives like masked language modeling (MLM) or next sentence prediction (NSP). The pre-trained models can then be fine-tuned on labeled text classification datasets to adapt them to specific tasks.

8) Transformer Variants - Various transformer variants have been developed, including BERT (Bidirectional Encoder Representations from

Transformers) [70] Figure 2.13 An overview of transformer learning for text classification, GPT (Generative Pre-trained Transformer) [72], RoBERTa (Robustly optimized BERT approach) [73], and XLNet (eXtreme Learning Transformer) [67][74]. These variants differ in architecture, pre-training objectives, and fine-tuning strategies but share the fundamental transformer architecture.

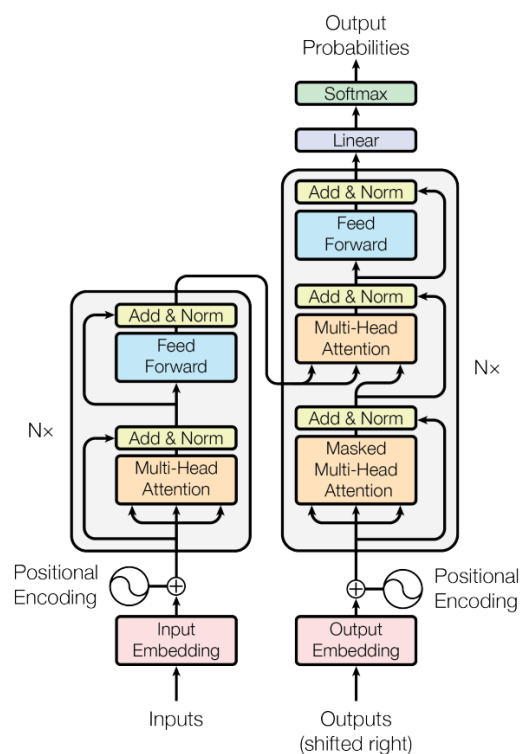


Figure 2.13 An overview of transformer learning for text classification

From: <https://blog.paperspace.com/transformers-text-classification/>

Transformer models excel at capturing long-range dependencies, modeling contextual information, and achieving state-of-the-art performance on various NLP tasks, including text classification. They require less manual feature engineering compared to traditional methods and can learn task-specific representations from raw text data. This approach has revolutionized text classification by leveraging self-attention mechanisms to capture contextual information effectively. Pre-trained transformer models, combined with fine-tuning on specific text classification tasks,

offer a powerful approach to achieving high performance in aspect-based text classification and other NLP tasks. Figure 2.6 presents a structure of transformer learning.

BERT is an example approach of transformer learning introduced by Google AI researchers in 2018. It is based on the transformer architecture, which is a deep learning model architecture that relies on self-attention mechanisms to capture global dependencies in input sequences. Transformers allow for parallel processing of tokens in the input sequence, making them highly efficient for NLP tasks. BERT is pre-trained on large-scale text corpora. Some of the notable BERT types include:

1. BERT Base: This is the original BERT model introduced by Google AI researchers. It consists of 12 transformer encoder layers and 110 million parameters.
2. BERT Large: This is a larger version of the BERT model with 24 transformer encoder layers and 340 million parameters. It provides higher capacity and potentially better

The sequence of processes involved in using BERT for text classification typically includes the following steps:

Step 1: Tokenization

The input text is tokenized into subword or word-level tokens using the BERT tokenizer. This tokenizer splits the text into tokens that are compatible with the vocabulary of the pre-trained BERT model. It also adds special tokens such as [CLS] (classification) and [SEP] (separator) to mark the beginning and end of the input sequence.

Step 2: Input Encoding

The tokenized input text is encoded into numerical representations suitable for input to the BERT model. Each token is mapped to its corresponding token ID based on the vocabulary of the pre-trained BERT model.

Step 3: Padding and Truncation

The tokenized sequences may have varying lengths, so they are padded with special padding tokens or truncated to a fixed length to fit the model's input size requirements. This ensures that all input sequences have the same length for batch processing.

Step 4: Input Representation

The padded or truncated token sequences are converted into dense vector representations using pre-trained word embeddings such as Word2Vec, GloVe, or BERT embeddings. BERT embeddings are context-aware and capture the contextual meaning of each token based on its surrounding context.

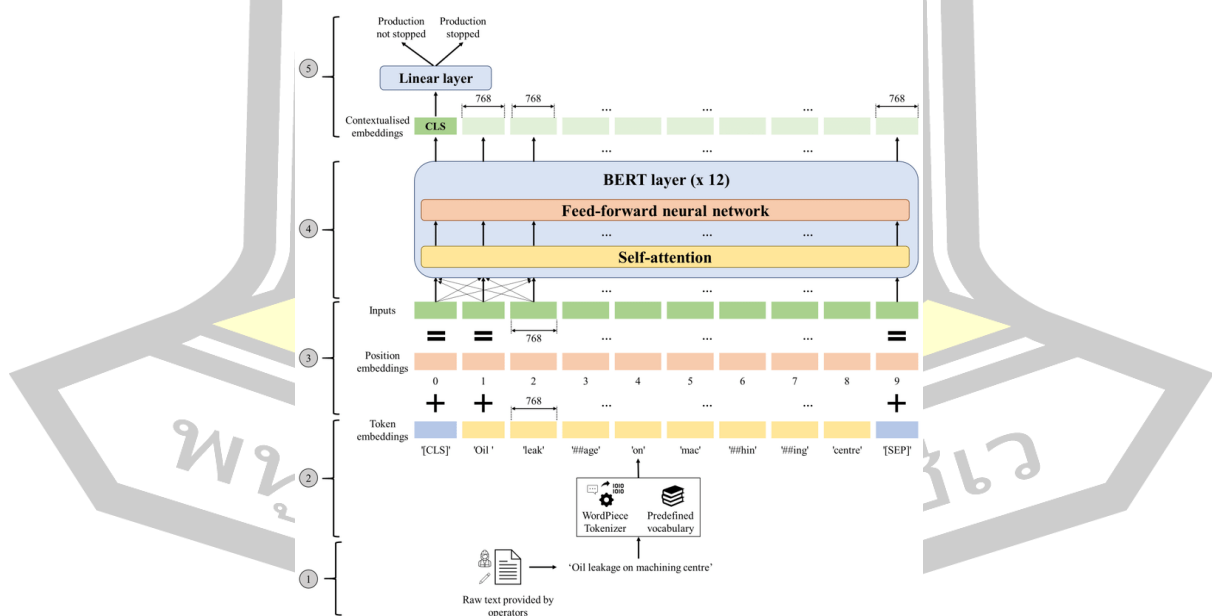


Figure 2.14 An overview of BERT for text classification

From: https://www.researchgate.net/figure/Example-of-a-trained-BERT-for-text-classification_fig1_353419108

Step 5: Special Tokens Addition

Special tokens like [CLS] (classification) and [SEP] (separator) are added to the beginning and end of the input sequence, respectively. The [CLS] token is used to represent the entire input sequence, and the [SEP] token separates different segments or sentences in the input.

Step 6: BERT Model Processing

The encoded input sequences are passed through the pre-trained BERT model, which consists of multiple transformer encoder layers. The BERT model generates contextual embeddings for each token in the input sequence, capturing its meaning based on its surrounding context.

Step 7: Classification Layer Addition

On top of the BERT model, a classification layer is added for the specific text classification task. This layer typically consists of one or more fully connected layers followed by a softmax activation function for multi-class classification or a sigmoid activation function for binary classification.

Step 8: Fine-tuning

The entire model, including the pre-trained BERT layers and the classification layer, is fine-tuned on a labeled text classification dataset using supervised learning. During fine-tuning, the model learns task-specific features and representations from the BERT embeddings to optimize performance on the classification task.

Step 9: Evaluation

After fine-tuning, the performance of the fine-tuned BERT model is evaluated on a separate validation or test set to assess its classification accuracy and other metrics. Adjustments to hyperparameters and model architecture may be made based on the evaluation results to improve performance.

2.3.5 Evaluation Metrics for Aspect-based Text Classification

1. Confusion Matrix

The confusion matrix, sometimes referred to as the likelihood matrix or the error matrix, is a fundamental component in the field of machine learning [75][76]. Confusion matrices serve as visual aids, primarily in the context of supervised learning, whereas in unsupervised learning they are commonly referred to as matching matrices. Evaluation mostly involves comparing the classification outcomes with the actual measured values, and the accuracy of the classification findings can be visually represented using a confusion matrix. The structure of the confusion matrix is typically shown as follows.

Each column in the confusion matrix corresponds to the expected class, and the sum of each column reflects the count of data predicted to belong to that class. Each row corresponds to the actual classification class of the data, and the total count of data in each row shows the number of occurrences in that class; the value in each column indicates the number of correctly predicted data belonging to that class.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Figure 2.15 Confusion Matrix

From: <https://encord.com/glossary/confusion-matrix/>

True Positive (TP): It refers to an outcome where the model correctly predicts the positive class. Simply speaking, it indicates that the model accurately identified an instance as belonging to a certain category when it indeed should have.

False Negative (FN): It refers to an outcome where the model incorrectly predicts the negative class for an instance that actually belongs to the positive class. It represents a mistake where the model fails to identify an instance as positive when it truly is, effectively missing a true positive case.

False Positive (FP): It refers to an outcome where the model incorrectly predicts the positive class for an instance that actually belongs to the negative class. This is essentially a type of error where the model has identified something as true or positive (according to the classification task) when it is not.

True Negative (TN): It refers to an outcome where the model correctly predicts the negative class for an instance that actually belongs to the negative class. This means that the model accurately identifies an instance as not belonging to the category (or categories) of interest.

2. Accuracy (Acc)

The metric of accuracy is often employed as a classification performance measure [77]. The accuracy of a model may be measured by calculating the ratio of correct identifications made by the model to the total number of samples. Typically, a model's quality improves as its accuracy increases. The formula can be written as (2.8).

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \quad (2.8)$$

3. Precision (P)

The precision ratio, sometimes referred to as the positive predictive value, is the proportion of samples classified as positive by the model that are truly positive [78].

Typically, a higher precision rate indicates a superior model. The formula can be written as (2.9).

$$P = \frac{TP}{TP + FP} \quad (2.9)$$

4. Recall (R)

Recall is a measure that quantifies the proportion of positive class samples correctly recognized by the model, relative to the total number of positive class samples [79]. Typically, higher recall indicates that the model accurately predicts more positive class samples, indicating a better performance of the model. The formula can be written as (2.10).

$$R = \frac{TP}{TP + FN} \quad (2.10)$$

5. The Receiver Operating Characteristic (ROC)

The ROC curve is a graphical representation used to evaluate the performance of a binary classifier system as its discrimination threshold is varied [80][81]. It plots two parameters:

- 1) True Positive Rate (TPR): Also known as sensitivity or recall, it measures the proportion of actual positives that are correctly identified by the model. It is calculated as $TPR = TP / (TP + FN)$, where TP is the number of true positives and FN is the number of false negatives.
- 2) False Positive Rate (FPR): It measures the proportion of actual negatives that are incorrectly identified as positives by the model. It is calculated as $FPR = FP / (FP + TN)$, where FP is the number of false positives and TN is the number of true negatives.

The ROC curve plots TPR against FPR at various threshold settings. The threshold refers to the probability (or some other measure) at which the classification decision changes from one class to another. Adjusting the threshold affects the classifier's sensitivity and specificity, and thus changes the FPR and TPR values.

In Figure 2.16, the ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.

6. The Area Under the Curve (AUC)

The AUC specifically refers to the area under the ROC curve, a popular evaluation metric used in binary classification to understand a model's diagnostic ability [60][82]. The AUC represents a probability measure of a classifier's ability to distinguish between the classes and is used to quantify the overall performance of a classification model. Key aspects of AUC are:

Value Range: The AUC ranges from 0 to 1, where an AUC of 1 indicates a perfect model that can completely distinguish between positive and negative classes. An AUC of 0.5 suggests a model with no discriminative ability, equivalent to random guessing. In practice, an AUC below 0.5 indicates a model performing worse than random guessing, but this situation typically leads to inverting the model's predictions to improve performance.

Interpretation: A higher AUC value means that the model has a better performance in distinguishing between the positive and negative classes across all possible thresholds. It integrates the model's performance across all classification thresholds, making it less sensitive to changes in the decision threshold than other metrics like accuracy.

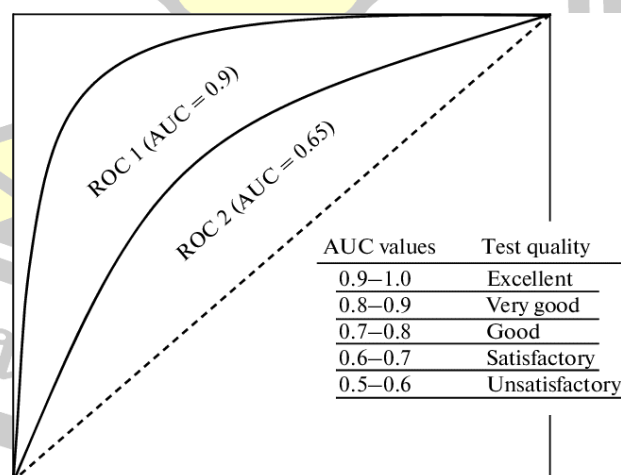


Figure 2.16 ROC and AUC

https://www.researchgate.net/figure/An-example-of-ROC-curves-with-good-AUC-09-and-satisfactory-AUC-065-parameters_fig2_276079439

2.4 Word2Vec

Word2Vec [83][84] is a commonly employed method in natural language processing (NLP) that effectively generates word embeddings, which are compact vector representations of words. These embeddings encapsulate semantic meanings, syntactic connections, and diverse language patterns derived from the contextual occurrences of words. Word2Vec models are trained by utilizing a vast collection of text, in which the model acquires knowledge about representations that position words with comparable meanings in close proximity within a multi-dimensional vector space. The method, which was developed by Tomas Mikolov and his colleagues at Google, is executed using two primary frameworks: (1) Continuous Bag of Words (CBOW) and (2) Skip-Gram. The CBOW model is generally used to predict a target word from a set of context words surrounding it, while the Skip-Gram model is generally used to predict context words from a target word.

2.4.1 Continuous Bag of Words (CBOW) [85][86]

CBOW model is one of the two architectures provided by the Word2Vec technique for generating word embeddings. In CBOW, the goal is to predict a target word based on a set of context words. This model treats context as a 'bag' of words, meaning the order of words does not significantly affect the prediction. The framework of CBOW involves several key components:

Input Layer: This consists of several parts, each corresponding to one of the context words. These parts are typically one-hot encoded vectors representing the words in the vocabulary.

Projection Layer: The one-hot encoded vectors are projected to a hidden layer but, unlike traditional neural networks, there is no activation function applied in this layer. The projection is a weighted sum of the context word vectors (i.e., the embeddings). In the case of CBOW, these vectors are averaged together.

Output Layer: The output is a single vector (the same size as the vocabulary) representing the probability distribution of the vocabulary words, indicating the likelihood of each word being the target word given the context.

Loss Function: The model typically uses a softmax function at the output layer to estimate probabilities, and the training process involves minimizing a loss function, such as cross-entropy, between the predicted probability distribution and the actual distribution (which is the one-hot encoded target word).

Figure 2.17 shows the architecture diagram for CBOW that would illustrate these components and the flow of data from the input layer (context words), through the projection layer (where embeddings are averaged), to the output layer (predicting the target word), and finally calculating the loss to adjust the embeddings during training.

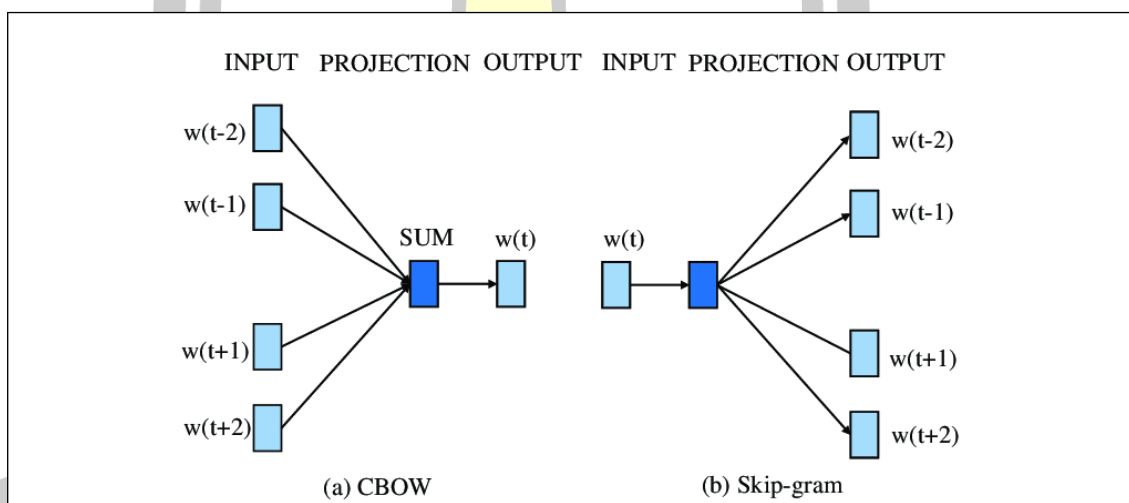


Figure 2.17 The architecture diagram for CBOW and Skip-gram

From: https://www.researchgate.net/figure/The-CBOW-and-Skip-gram-architecture-of-word2vec-16_fig1_329768544

The diagram shown in Figure 2.17 (a) illustrates the CBOW model framework for Word2Vec. It shows the flow from multiple input nodes representing context words (as one-hot encoded vectors) through the projection layer where these vectors

are averaged, to the output layer. At the output layer, a softmax function predicts the target word from the vocabulary. The process also involves using a loss function for training, which helps adjust the word embeddings based on the prediction accuracy. This visual representation helps in understanding how CBOW predicts a target word based on the context provided by surrounding words.

Following example of the CBOW for the sentence “The quick brown fox jumps over the lazy dog.” can be illustrated as follows.

If using the CBOW model to predict the word “fox” based on the context words “quick,” “brown,” “jumps,” and “over.” Here’s a step-by-step example of how this might be represented and processed:

Step 1: Pre-processing - First, the sentence would be tokenized into individual words, and possibly, each word would be converted to a lower case or undergo other preprocessing steps to normalize the text.

Step 2: One-hot Encoding - Each word in the vocabulary is represented as a one-hot encoded vector. For simplicity, assume our vocabulary only consists of the unique words in the sentence, and each word is assigned an index. For example:

“the”: [1, 0, 0, 0, 0, 0, 0, 0, 0]

“quick”: [0, 1, 0, 0, 0, 0, 0, 0, 0]

“brown”: [0, 0, 1, 0, 0, 0, 0, 0, 0]

“fox”: [0, 0, 0, 1, 0, 0, 0, 0, 0]

“over”: [0, 0, 0, 0, 1, 0, 0, 0, 0]

(and so on for each word)

Step 3: Input to the Model - For predicting “fox,” it takes the one-hot vectors of “quick,” “brown,” “jumps,” and “over” as input. Since it’s CBOW, it averages these vectors. In a more complex model, these would be embedded vectors, but for simplicity, imagine averaging their one-hot representations.

Step 4: Prediction - The CBOW model processes the averaged input through its neural network layers to produce a prediction. The output is a probability distribution

over the vocabulary, indicating the likelihood of each word being the correct prediction.

Step 5: Output - The model outputs the word “fox” as the highest probability word, given the context words “quick,” “brown,” “jumps,” and “over.”

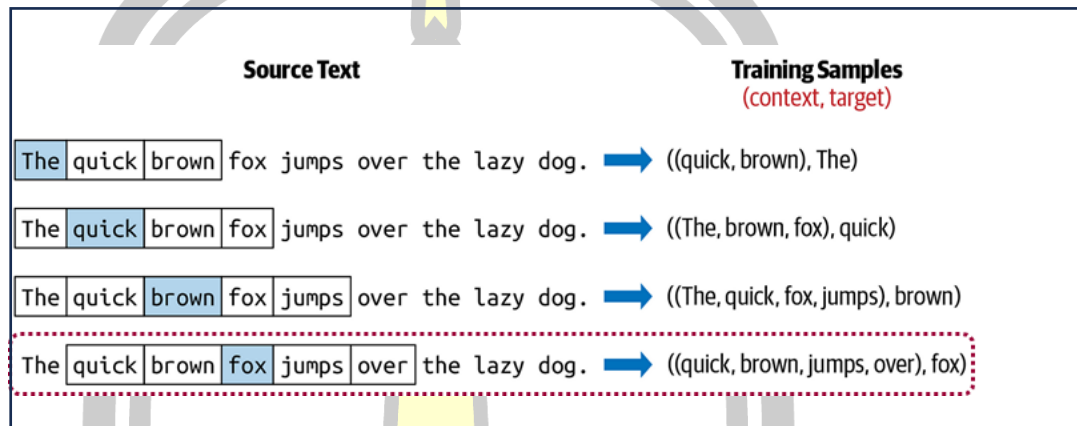


Figure 2.18 An example of using the CBOW model to predict the word “fox”

From: <https://medium.com/@evertongomede/understanding-the-continuous-bag-of-words-cbow-model-586c5f60cb0d>

2.4.2 Skip-Gram [87][88]

The Skip-Gram model is the second architecture provided by the Word2Vec technique for generating word embeddings. Unlike the CBOW model, which predicts a target word from a set of context words, the Skip-Gram model works in reverse: it uses a target word to predict the surrounding context words. This model is particularly useful for handling large corpora and capturing rare words or phrases. The framework of Skip-Gram involves several components:

Input Layer: This consists of a one-hot encoded vector representing the target word.

Projection Layer: Similar to CBOW, the input vector is projected to a hidden layer without applying any activation function. However, in Skip-Gram, there's a direct mapping since there's only one word being processed.

Output Layer: Unlike CBOW, the output layer in the Skip-Gram model consists of multiple parts, each corresponding to a context word that the model attempts to predict. The output is a set of vectors (each the size of the vocabulary) representing the probability distributions of the vocabulary words, indicating the likelihood of each word being in the context of the target word.

Loss Function: The model uses a softmax function at each output part to estimate probabilities. The training process involves minimizing a loss function, such as cross-entropy, between the predicted probability distribution and the actual distribution of the context words.

Figure 2.17 shows the architecture diagram for Skip-Gram that would illustrate these components, showing the flow from the input layer (target word), through the projection layer, to the multiple output layers (predicting context words), and finally, calculating the loss to adjust the embeddings during training.

The diagram shown in Figure 2.17 (b) illustrates the Skip-Gram model framework for Word2Vec. It demonstrates the process from the single input node representing the target word (as a one-hot encoded vector) through the projection layer to multiple output nodes. These output nodes use softmax functions to predict the context words from the vocabulary based on the target word. The diagram also highlights the use of a loss function for training, which adjusts the word embeddings to improve the accuracy of predicting context words. This visual representation helps in understanding how Skip-Gram predicts multiple context words from a single target word, effectively capturing word relationships and semantics within a given corpus.

Let's provide an illustration of how the Skip-Gram model from Word2Vec may function by analyzing an example sentence "*The quick brown fox jumps over the lazy dog.*" It will use the Skip-Gram model to predict the context words based on the target word "*fox.*" Here's a step-by-step example:

Step 1: Pre- processing - The sentence is tokenized into individual words, and various preprocessing steps are applied as necessary (e.g., converting to lowercase, removing punctuation).

Step 2: One-hot Encoding - Each word in the vocabulary is represented as a one-hot encoded vector. Assuming a simplified vocabulary from the sentence, where each word is uniquely represented, for example:

“the”: [1, 0, 0, 0, 0, 0, 0, 0, 0]

“quick”: [0, 1, 0, 0, 0, 0, 0, 0, 0]

“brown”: [0, 0, 1, 0, 0, 0, 0, 0, 0]

“fox”: [0, 0, 0, 1, 0, 0, 0, 0, 0]

“over”: [0, 0, 0, 0, 1, 0, 0, 0, 0]

(and so on for each word)

Step 3: Input to the Model - The one-hot vector for “fox” is used as the input to the Skip-Gram model.

Step 4: Prediction - The Skip-Gram model processes the input through its neural network layers to predict the context words. The output is several probability distributions over the vocabulary, each corresponding to a position in the context around “fox” (e.g., one position before, one position after).

Step 5: Output - Based on the context size (let’s assume it’s 2 for simplicity, meaning two words before and two words after the target word), the model might predict high probabilities for:

- “brown” and “quick” as likely predecessors
- “jumps” and “over” as likely successors

Imagine the word “fox” is at the center, and the model, through its training, has learned to associate certain probabilities with words that are likely to appear in its vicinity. So, given “fox,” it strengthens the association with “quick,” “brown,” “jumps,” and “over” based on its understanding from the training corpus.

This simplified example demonstrates the basic principle of how the Skip-Gram model aims to predict context words for a given target word, capturing the relationships between a word and its surrounding words. Through training on large text corpora, the model learns high-quality word embeddings that reflect these semantic and syntactic patterns.

2.5 K-means Clustering Approach

K-means clustering [89][90] is a popular unsupervised machine learning algorithm used for partitioning a dataset into a set of k groups (or clusters), where k is a predefined number. The goal is to minimize the variance within each cluster and maximize the variance between clusters. It's commonly used in data analysis for pattern recognition, image analysis, information retrieval, and various other applications. The approach of *K*-means clustering can be described as follows.

1. Initialization: This algorithm starts by selecting k initial centroids, where k is the number of clusters you want to identify in your dataset. These centroids can be chosen randomly or by using more sophisticated methods to improve convergence.

2. Assignment Step: It is to assign each data point to the nearest centroid, based on the distance between the data point and the centroid. The Euclidean distance is commonly used, but other distance measures can also be applied depending on the nature of the data. The formula for Euclidean distance between two points P and Q in an n -dimensional space, where P and Q are defined by their coordinates $P = (p_1, p_2, \dots, p_n)$ and $Q = (q_1, q_2, \dots, q_n)$, respectively is given by:

$$d(P, Q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \quad (2.11)$$

This can be simplified to:

$$d(P, Q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.12)$$

Exempl for a two-dimensional space, the Euclidean distance between two points $P(x_1, y_1)$ and $Q(x_2, y_2)$ is calculated as:

$$d(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.13)$$

And for a three-dimensional space, for points $P(x_1, y_1, z_1)$ and $Q(x_2, y_2, z_2)$ is calculated as:

$$d(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (2.14)$$

The Euclidean distance formula thus provides a straightforward and intuitive way to measure the distance between points in any dimensional space, serving as a foundational metric in various areas of mathematics, physics, and computer science.

3. Update Step: Once all data points have been assigned to clusters, recalculate the centroids by taking the mean of all points in each cluster.

4. Iteration: It is to repeat the assignment and update steps until the centroids no longer change significantly, indicating that the algorithm has converged, or until a specified number of iterations is reached.

2.6 Related Work

Ganganwar, V. et al. [91] note that there has been limited research on the extraction of hidden traits, and while detecting implicit aspects presents significant challenges, these aspects are crucial for comprehensive sentiment analysis. In their study examining papers from multiple databases between 2014-2019, they provided an overview of various methods for identifying implicit features, including rule-based approaches, statistical methods, and machine learning techniques. The authors' limitations and suggestions for future research were carefully analyzed while conducting the research classification, incorporating their proposed methodology for implicit aspect extraction. Their analysis revealed several technical challenges in implicit aspect extraction, such as multiple aspect detection and contextual ambiguity, thus identifying promising directions for future research.

Majumder, N. et al. [92] provided an in-depth study of sentiment analysis's implicit and explicit aspect extraction techniques. This paper proposes a method to enhance Aspect-Level Sentiment Analysis (ALSA) by leveraging knowledge from Aspect Extraction (AE), using a Bidirectional Gated Recurrent Unit (BiGRU) and Conditional Random Field (CRF) for AE and transferring the learned word embeddings to ALSA models. Experiments were conducted on the SemEval-2014 Task 4 dataset across the Laptop and Restaurant domains, evaluating three baseline ALSA models (TC-LSTM, ATAE, IAN) and their transfer learning variants (TC-LSTM-T, ATAE-T, IAN-T). Results demonstrate that transfer learning significantly improves ALSA performance (average gain of 1.53%), with cross-domain transfer also showing promising results. This work is valuable for my research as it validates the effectiveness of cross-task knowledge transfer and provides insights into multi-domain adaptability, while the BiGRU-CRF framework for AE offers a technical reference for designing my aspect extraction module.

Schouten, K. et al. [93] discovered additional evaluations that are closely associated with ABSA in South Africa, including aspect-level sentiment analysis (SA), feature extraction in SA, opinion types in SA, aspect extraction in SA, and implicit aspect extraction in SA. They did a thorough examination of the current advancements in the field, revealing significant improvement in identifying all the targets. The targets might refer to either an entity or a specific component of that entity, together with the related emotion. The study focuses on ABSA and seeks to determine and summarize opinions or sentiments on entities or features specified in papers. This study differed from others in that its objective was to ascertain the attitudes associated with entities referenced in the papers, while their research mostly concentrated on methodologies for extracting implicit elements.

Cruz, I. et al. [94] introduced a novel method for extracting Implicit Aspect Indicators (IAI) in Aspect-Based Opinion Mining (ABOM). They developed a supervised approach using Conditional Random Fields (CRF) that integrates multiple

feature types including word features, character n-grams, context information, and class sequences. Their main contribution includes creating the first publicly available manually annotated corpus for IAI extraction, containing 314 Amazon product reviews across 5 electronic products. Through rigorous evaluation using 10-fold cross-validation, comparing it against three baselines: sentiment word labeling, a Naive Bayes classifier, and a Hidden Markov Model. The results show that the CRF-based approach achieves higher precision and recall. This work is particularly relevant to our research as it provides insights into feature engineering for implicit aspect detection.

In Li, Z. et al. [95], proposed a novel Supervised Contrastive Pre-training (SCAPT) approach to improve aspect-based sentiment analysis by better capturing implicit sentiment. The authors identified that about 30% of product reviews convey sentiment without explicit opinion words, which they term implicit sentiment. To address this challenge, they developed a pre-training method using large-scale sentiment-annotated corpora from in-domain sources (1.56M restaurant reviews from Yelp and 0.51M laptop reviews from Amazon). Their approach incorporates three key components: supervised contrastive learning to align representations of similar sentiment expressions, masked aspect prediction, and review reconstruction. On the SemEval-2014 benchmarks, SCAPT achieved state-of-the-art performance, with particularly strong improvements on implicit sentiment expressions (4.49% and 4.60% gains over previous methods on restaurant and laptop domains respectively). This work demonstrates the effectiveness of leveraging large-scale pre-training with contrastive learning to capture both explicit and implicit sentiment expressions.

Li, J. et al. [96] proposed SK2, a unified framework that integrates both implicit sentiment knowledge and explicit syntax knowledge for aspect-based sentiment analysis (ABSA). Their approach consists of two key components: (1) A self-supervised pre-training procedure incorporating four tasks (conjunctive words prediction, sentiment-word polarity prediction, attribute nouns prediction, and

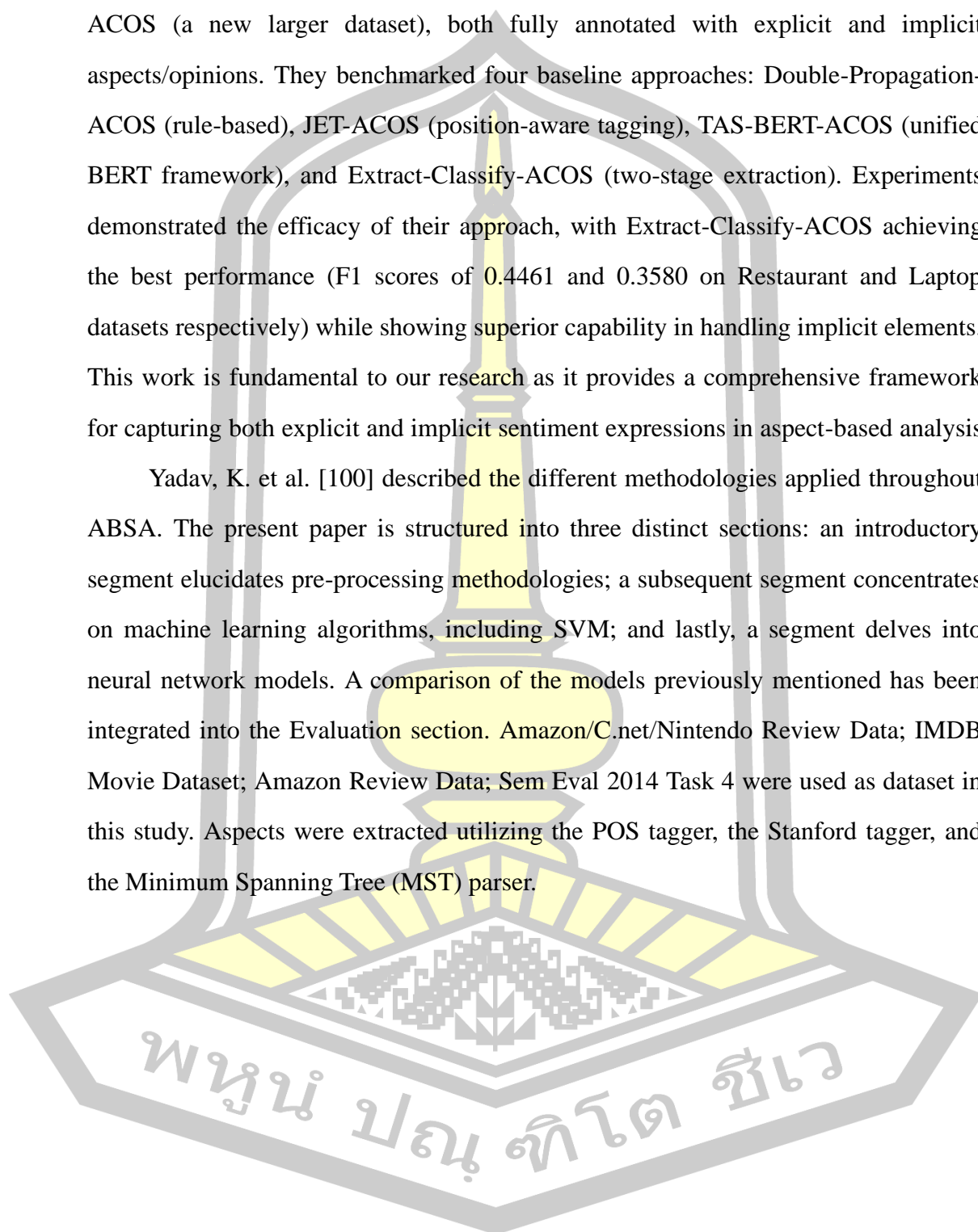
sentiment-oriented masked language modeling) to capture implicit sentiment knowledge; and (2) A sparse relational graph attention network (SR-GAT) with aspect-oriented dependency trees to encode explicit syntax knowledge. Evaluated on all seven ABSA tasks across three benchmarks (SemEval-2014/2015/2016), SK2 achieved significant improvements over previous methods, with average F1 score gains of 2.89%, 1.87%, and 5.43% on aspect extraction, opinion extraction, and aspect sentiment classification respectively. This work demonstrates the effectiveness of combining both sentiment and syntactic knowledge in a unified framework for comprehensive ABSA tasks.

Benarafa, H. et. al. [97] introduced a technique to enhance the performance of K-Nearest Neighbors (KNN) for implicit aspect identification (IAI) tasks. They suggested a revision to the calculation of KNN distance using WordNet semantic relations, which would assist in IAI tasks. To conduct a comprehensive empirical assessment, they performed experiments on two datasets consisting of electronic products and restaurant reviews. They evaluated the impact of our approach using three criteria: the KNN distance utilized for similarity computation, the number of nearest neighbors (K), and the behavior of KNN in relation to Overfitting and Underfitting. The experimental findings demonstrated that they approach improved the performance of KNN and its ability to address overfitting and underfitting issues in implicit aspect identification.

Cai, H. et al. [99] pioneered the Aspect-Category-Opinion-Sentiment (ACOS) Quadruple Extraction task to systematically address a critical challenge in Aspect-Based Sentiment Analysis - the handling of implicit aspects and opinions. Their analysis revealed that around 44% of product reviews contain implicit aspects or opinions, highlighting the significance of this previously overlooked issue. To address this, they developed a framework to extract quadruples consisting of aspect terms, aspect categories, opinion terms, and sentiment polarities, with NULL values specifically designed to handle implicit elements. The authors constructed two

comprehensive datasets: Restaurant-ACOS (extending SemEval 2016) and Laptop-ACOS (a new larger dataset), both fully annotated with explicit and implicit aspects/opinions. They benchmarked four baseline approaches: Double-Propagation-ACOS (rule-based), JET-ACOS (position-aware tagging), TAS-BERT-ACOS (unified BERT framework), and Extract-Classify-ACOS (two-stage extraction). Experiments demonstrated the efficacy of their approach, with Extract-Classify-ACOS achieving the best performance (F1 scores of 0.4461 and 0.3580 on Restaurant and Laptop datasets respectively) while showing superior capability in handling implicit elements. This work is fundamental to our research as it provides a comprehensive framework for capturing both explicit and implicit sentiment expressions in aspect-based analysis.

Yadav, K. et al. [100] described the different methodologies applied throughout ABSA. The present paper is structured into three distinct sections: an introductory segment elucidates pre-processing methodologies; a subsequent segment concentrates on machine learning algorithms, including SVM; and lastly, a segment delves into neural network models. A comparison of the models previously mentioned has been integrated into the Evaluation section. Amazon/C.net/Nintendo Review Data; IMDB Movie Dataset; Amazon Review Data; Sem Eval 2014 Task 4 were used as dataset in this study. Aspects were extracted utilizing the POS tagger, the Stanford tagger, and the Minimum Spanning Tree (MST) parser.



CHAPTER III

RESEARCH METHODOLOGY

This section outlines the proposed framework that will be utilized for the purpose of this study. The details of the datasets and specific task inside the framework can be explained in the following manner.

3.1 Datasets

Dataset is fundamental to the training and evaluation of machine learning and deep learning models. In aspect-level sentiment analysis tasks, high-quality datasets should contain not only user review texts but also aspect annotations and corresponding sentiment polarity labels. These annotated data enable models to learn complex relationships between semantic features, aspect expressions, and sentiment orientations in the text. Standardized datasets also provide unified evaluation benchmarks for comparing different research outcomes, ensuring the comparability and reproducibility of research results.

The collection and curation of datasets are crucial for aspect-level sentiment analysis research. Firstly, accurate data annotation directly impacts the model's learning effectiveness and prediction performance. Particularly for implicit aspect sentiment analysis, since aspect terms do not explicitly appear in the text, the annotation process needs to consider contextual semantics and domain knowledge, which increases the complexity of dataset construction. Secondly, the scale and diversity of datasets determine the model's generalization ability.

In implicit aspect sentiment analysis research, scholars have employed several public datasets for experimental verification. The restaurant and laptop domain datasets provided by SemEval 2014 Task 4 have been widely used, containing numerous implicit aspect expressions Wang, Y. et al., [58]. Xiang, Z. et al. [101]

constructed a dataset containing 2000 hotel reviews, specifically annotated with implicit aspects and their sentiment polarity. Xu, B. et al. [67] utilized product review datasets from Amazon, focusing on implicit aspect identification. Additionally, Meng, W. et al. [56] established a cross-domain Chinese review dataset, covering implicit aspect expressions in various domains such as dining and hotels. The construction and application of these datasets have advanced implicit aspect sentiment analysis research, providing crucial support for model improvements.

Our dataset was sourced from TripAdvisor, three linguists then assist in determining which specific hotel aspect (i.e., staff hotel service, hotel cleanliness, value for money, and hotel location) is featured in those sentences and assist in determining the sentiment polarity of those sentences. consisting of English-written reviews with a 5-star rating system. Reviews rated 4 or 5 were designated as positive class, while those with ratings of 1 or 2 were categorized as negative class. Reviews with a rating of 3 were omitted as they were presumed to reflect neutral sentiments.

In our dataset, there are various numbers of sentences in each hotel review. Some hotel reviews contain numerous sentences describing multiple aspects of the hotel, while others contain only a few words. After our linguists segmented the sentences by aspects, most sentences contain fewer than 40 words. The aspects examined in this study encompassed staff service, cleanliness, value, and location convenience. The summary of our dataset can be described in Table 3-1.

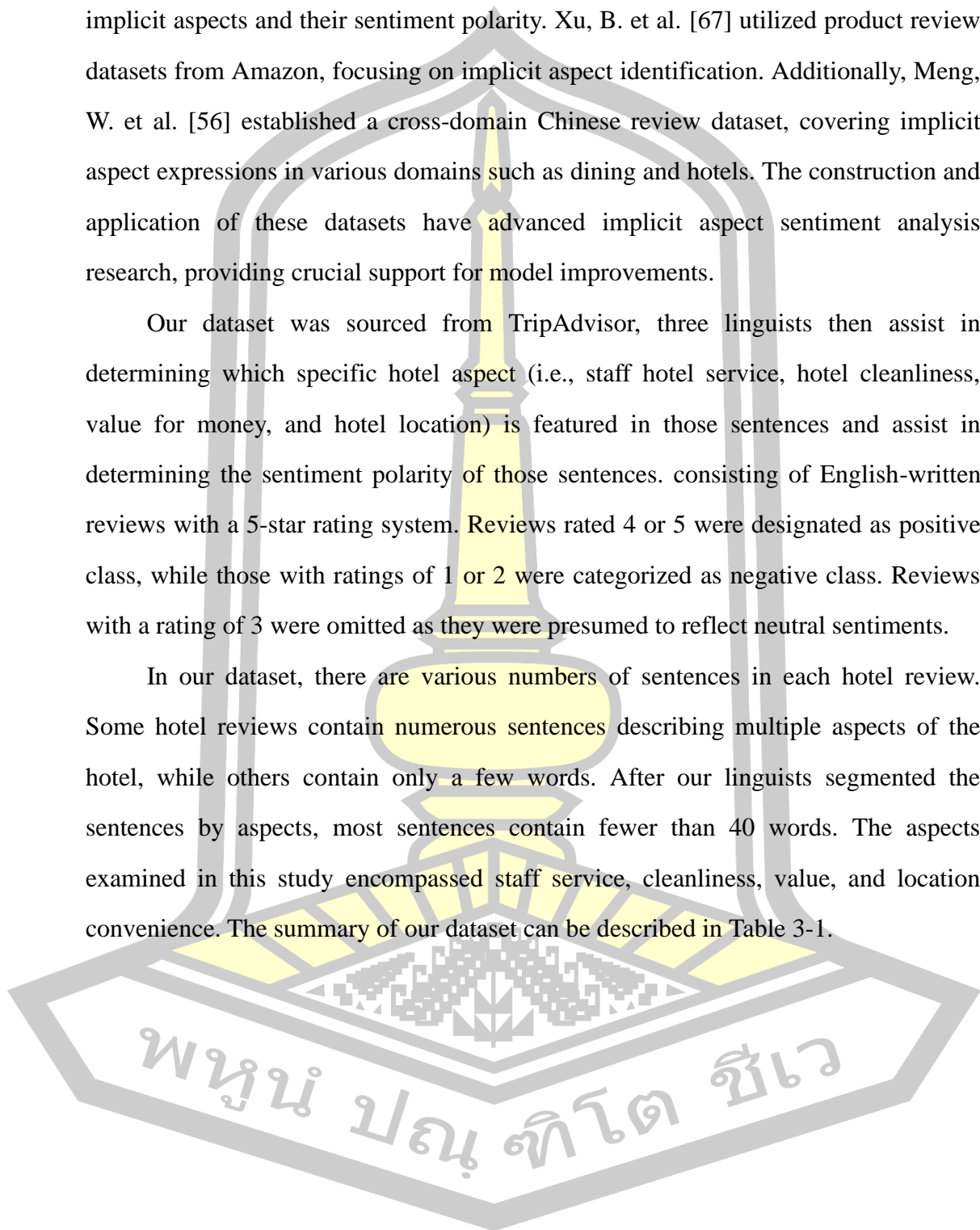


Table 3-1 Summarization of the dataset

Hotel Aspects	Current Number of Relevant Sentences			
	Explicit	Implicit	positive	negative
staff service	397	153	380	170
cleanliness	385	165	290	260
value price	300	150	215	235
convenience of location	333	117	341	109

3.2 The Framework

This section describes a proposed framework overview in this study (See in Figure 3.1).

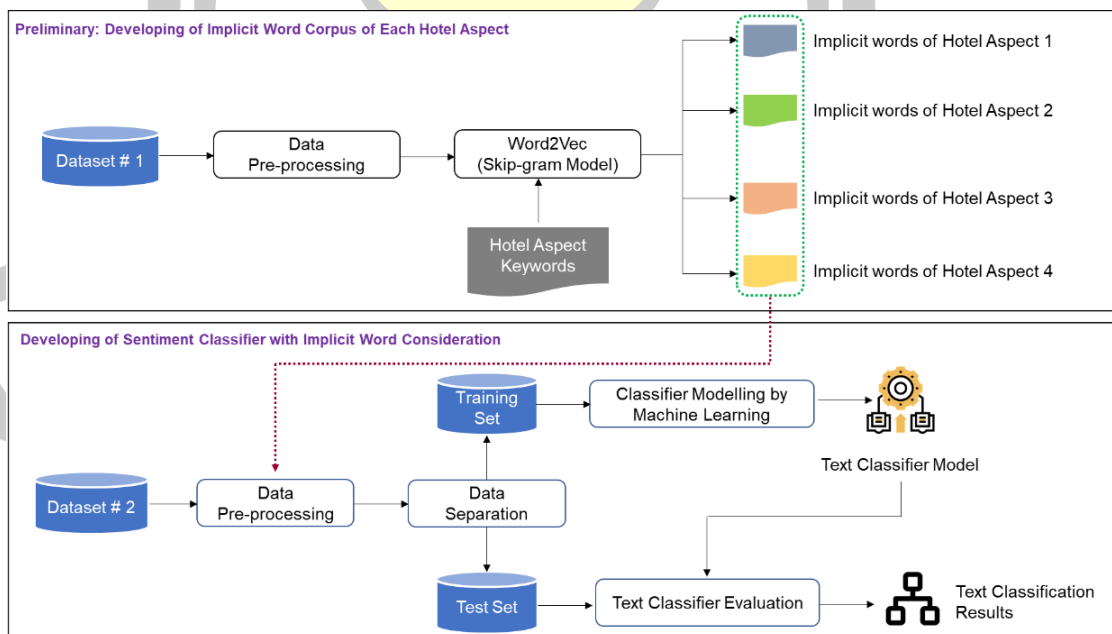


Figure 3.1 The Framework

Figure 3.1 depicts the proposed framework in this study. The structure has two primary components. Initially, the goal is to create a preliminary corpus that captures implicit word associations for every aspect of the hotel. Secondly, the objective is to develop a binary aspect-based sentiment classifier for every aspect of the hotel.

3.3 Preliminary: Developing Implicit Word Corpus of Each Hotel Aspect

The Skip-gram model, part of the Word2Vec family of models, is designed to predict the context for a given target word. It works by taking a target word as input and trying to predict the surrounding words (context) within a specific window size from the target word. This characteristic makes Skip-gram particularly useful for finding words that frequently appear in similar contexts as the given keywords, effectively identifying relevant words related to those specific keywords. Here is a common methodology for utilizing Skip-gram to identify relevant words associated with specified keywords:

Step 1: Pre-processing

The pre-processing phase is fundamental to ensuring the quality and reliability of the word embedding model. This study utilizes dataset, which consists of domain-specific hotel reviews, to establish a robust corpus for training. The pre-processing is designed to optimize the textual data for subsequent semantic analysis while preserving domain-specific information.

The pre-processing workflow comprises several sequential stages, each serving a specific purpose in data refinement:

1. Text Cleaning-- Text Cleaning and Standardization constitute the initial phase of data preparation. This stage addresses the heterogeneous nature of user-generated content by implementing comprehensive text normalization protocols. The process systematically handles various text anomalies, including irregular encodings, extraneous HTML elements, and inconsistent character representations. Additionally, domain-specific expressions and numerical representations undergo standardization to ensure consistency throughout the corpus.

2. Linguistic Processing- Linguistic Processing forms the second phase, focusing on the structural and grammatical aspects of the text. This stage employs natural language processing techniques to decompose the text into its constituent linguistic units while preserving semantic relationships. The process maintains contextual integrity through careful sentence boundary detection and implements sophisticated tokenization algorithms that account for domain-specific linguistic patterns. The preservation of semantic significance guides the case normalization process, ensuring that meaningful distinctions in capitalization patterns are retained where semantically relevant.

3. Semantic Refinement- Semantic Optimization represents the final phase, concentrating on enhancing the semantic quality of the processed text. This stage implements context-aware stopword removal, utilizing specially curated stopword lists that account for the unique characteristics of hospitality domain discourse. The implementation of lemmatization techniques reduces morphological variants to their base forms while maintaining semantic distinctiveness. Part-of-speech tagging provides additional linguistic context, facilitating more accurate word sense disambiguation. Throughout this phase, particular attention is paid to preserving domain-specific terminology that carries significant semantic weight in the hospitality context.

The quality and relevance of the corpus significantly influence the model's capability to identify meaningful and relevant word associations. Therefore, particular attention is paid to preserving domain-specific terms and phrases that carry semantic weight in the hospitality context.

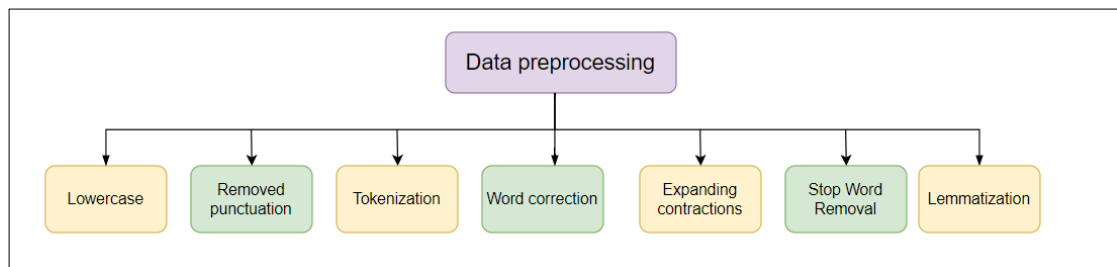


Figure 3.2 Sequential Steps in Text Pre-processing Framework

Step 2: Skip-gram Model Implementation

The Skip-gram model implementation builds upon the foundational theoretical framework established by Mikolov et al. [87]. This architectural choice is predicated on the model's demonstrated capacity to capture fine-grained semantic relationships and contextual associations, particularly within domain-specific corpora. The implementation architecture emphasizes both semantic precision and computational efficiency.

1. Model Architecture - The core architecture is designed to optimize the semantic representation capabilities while maintaining computational feasibility. The model employs a dense vector space representation with carefully calibrated parameters:

The embedding dimension is set to 300, a choice that balances representational capacity with computational efficiency. This dimensionality provides sufficient degrees of freedom to capture subtle semantic nuances while avoiding the curse of dimensionality. The context window extends to 5 words on either side of the target word, enabling the model to capture both immediate syntactic relationships and broader semantic associations.

The selection of 300 dimensions for the embedding space represents a carefully optimized choice based on both theoretical and empirical considerations. This dimensionality provides sufficient representational capacity to capture fine-grained semantic relationships while avoiding the computational overhead and potential noise

introduction of higher dimensions. Experimental analysis shows that 300 dimensions achieve optimal performance across multiple metrics:

- 1) Semantic Resolution: Provides sufficient granularity to distinguish subtle meaning differences
- 2) Training Stability: Enables consistent convergence during the training process
- 3) Computational Efficiency: Maintains reasonable memory requirements and processing speed
- 4) Information Density: Optimally balances information capture and noise reduction

Comparative testing with alternative dimensionalities (100-500) demonstrated that 300 dimensions represent an optimal point where. Lower dimensions (100-200) showed insufficient capacity to capture complex semantic relationships. Higher dimensions (400+) produced minimal performance improvements while significantly increasing computational costs.

2. Training Framework - The training methodology incorporates several key optimizations:

- 1) Learning Rate Optimization
 - Initial learning rate: 0.025
 - Dynamic adjustment: Linear decay schedule
 - Adaptive stepping: Gradual reduction based on training progress
- 2) Sampling Strategy
 - Negative sampling ratio: 5 negative samples per positive instance
 - Subsampling threshold: $1e-3$ for frequent term regulation
 - Minimum frequency threshold: 5 occurrences
- 3) Training Dynamics
 - Epoch configuration: 5 complete passes over the corpus
 - Batch processing: 128-word batches

- Convergence monitoring: Loss trajectory analysis

The implementation incorporates sophisticated error handling and monitoring mechanisms to ensure training stability and convergence. The parameter configuration reflects empirical optimization through systematic experimentation, with each parameter carefully tuned to maximize model performance while maintaining computational efficiency.

Step 3: Semantic Word Identification Methodology

This section presents a methodological framework for identifying semantically relevant words using the Skip-gram model. The framework consists of two principal components: a query mechanism and an interpretation framework, designed to enable systematic semantic analysis in vector space.

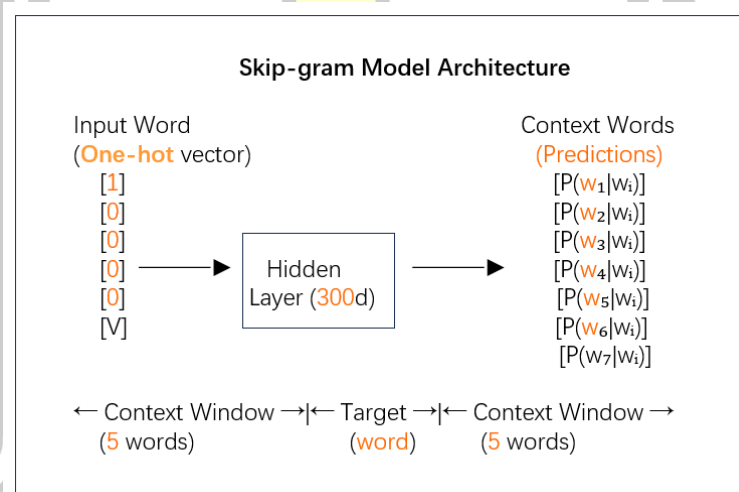


Figure 3.3 kip-gram Model Architecture

A. Query Mechanism Design

The query mechanism is designed to retrieve semantically related terms from a trained Skip-gram model. It operates through the following steps:

1. Vector Space Transformation

The Skip-gram model's learned weight matrix transforms input keywords into dense vector representations. The transformation is defined as:

$$v(w) = W \cdot x(w) \quad ($$

3.1)

where $v(w)$ represents the resultant word vector, W denotes the learned weight matrix, and $x(w)$ is the one-hot encoded input vector.

2. Similarity Computation

Semantic similarity between word vectors is calculated using cosine similarity, which measures the cosine of the angle between two vectors in the vector space:

$$\text{similarity}(w_1, w_2) = \cos \theta = \frac{v(w_1) \cdot v(w_2)}{\|v(w_1)\| \|v(w_2)\|} \quad (3.2)$$

This metric quantifies the degree of semantic relatedness between words.

B. Interpretation Framework

The interpretation framework provides a structured approach to analyze and validate the outputs of the Skip-gram model. It consists of two key components:

1. Similarity Assessment

- Quantitative evaluation of vector space relationships.
- Relative ranking of semantic similarities.
- Application of a threshold-based filtering.

2. Contextual Analysis

- Categorization of semantic relationships.
- Context-based validation.
- Domain-specific interpretation rules.

The query process is implemented through the following algorithmic framework:

C. Methodological Considerations

The effectiveness of the framework depends on several critical factors:

1. Query Design Principles

- Preprocessing of input terms (e.g., tokenization, lemmatization).

- Normalization of vector space for consistent similarity computation.
- Handling of multi-term queries and out-of-vocabulary terms.

2. Interpretation Guidelines

- Standardization of similarity scores for cross-domain comparability.
- Context-aware filtering of results to ensure relevance.
- Integration of domain knowledge for improved semantic understanding.
- Validation of relationships using external linguistic resources or expert input.

This methodology establishes a systematic approach to semantic word identification using the Skip-gram model. The framework integrates query mechanisms with interpretation guidelines, providing a foundation for robust semantic analysis. Detailed experimental evaluation and parameter optimization will be presented in Chapter 4.

3.4 Binary Aspect-based Sentiment Classifier Development

In this stage, the datasets are utilized to develop a binary aspect-based sentiment classifier. Developing a binary aspect-based sentiment classifier involves a more nuanced approach than general sentiment analysis. This type of classifier not only determines the sentiment (positive or negative) of a given text but also associates it with specific aspects or features of a hotel, i.e. staff service, cleanliness, value price, and convenience of location.

3.4.1 Text Pre-processing and Representation

Text pre-processing and representation are crucial steps in developing a binary aspect-based sentiment classifier. These steps transform raw text into a format that machine learning models can understand and use to learn patterns associated with sentiments towards specific aspects. The goal of text pre-processing is to clean and normalize the text data. This can help improve the model's performance by reducing noise and focusing on the meaningful parts of the text. Figure 3.4 presents the overall architecture of the proposed framework. This framework is specifically designed for

binary aspect-based sentiment classification, which determines both the sentiment polarity (positive/negative) and its association with specific hotel aspects (staff service, cleanliness, value price, and convenience of location).

The framework consists of three main components:

- 1) Text Normalization: Standardize raw text data.
- 2) Aspect-specific Processing: Identification and extraction of aspect-related contextual information.
- 3) Feature Representation: Construction of feature vectors for machine learning models.

1. Text Normalization (Linguistic Preprocessing)

The linguistic preprocessing pipeline consists of five core operations:

- Case Normalization
 - Converting all text to lowercase to ensure consistent token matching.
 - Example: “Great Service” → “great service”.
- Noise Removal
 - Eliminating punctuation, special characters, and irrelevant symbols using regular expressions.
 - Example: “Great service!” → “great service”
- Tokenization
 - Segmenting text into individual tokens using NLTK's punkt tokenizer.
 - Example: “great service” → [“great”, “service”].
- Stopword Elimination
 - Removing common words using NLTK’s stopword list.
 - Example: “the service was great” → [“service”, “great”]
- Lemmatization
 - Reducing words to their base form while preserving contextual meaning.
 - Example: “services” → “service”

To pre-process hotel reviews, we utilized the NLTK for tokenization, removing stopwords, and stemming, as well as regular expressions (re) for removing punctuation and special characters.

First, to ensure that NLTK is installed, we need to download NLTK using pip and download the stopwords and punkt tokenizer datasets using NLTK's download utility. Command for doing this can be shown Figure 3.4

```
pip install nltk
python -m nltk.downloader stopwords
python -m nltk.downloader punktmodel.fit([word_target, word_context],
np.array(labels), epochs=1000, verbose=2)
```

Figure 3.4 Instructions for downloading NLTK by pip

2. Aspect-specific Processing

To enable aspect-based sentiment analysis, the framework incorporates the following steps:

- 1) Aspect Term Identification: Detecting terms related to specific hotel aspects (e.g., “staff,” “cleanliness”) using domain-specific lexicons or rule-based methods.
- 2) Context Window Extraction: Capturing the surrounding text (e.g., ± 5 words) to provide context for sentiment analysis.
- 3) Aspect-Sentiment Association: Linking identified aspects with their corresponding sentiment expressions (e.g., “friendly staff” \rightarrow positive sentiment).

This processing step ensures that sentiment analysis is performed at a granular level, capturing the relationship between specific aspects and their associated sentiments.

3. Feature Representation

The feature representation framework implements a sophisticated approach to transform preprocessed textual data into numerical vectors that capture both statistical

significance and semantic relationships. At its core, the framework employs Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, augmented with aspect-specific feature engineering to enhance the representation's discriminative power.

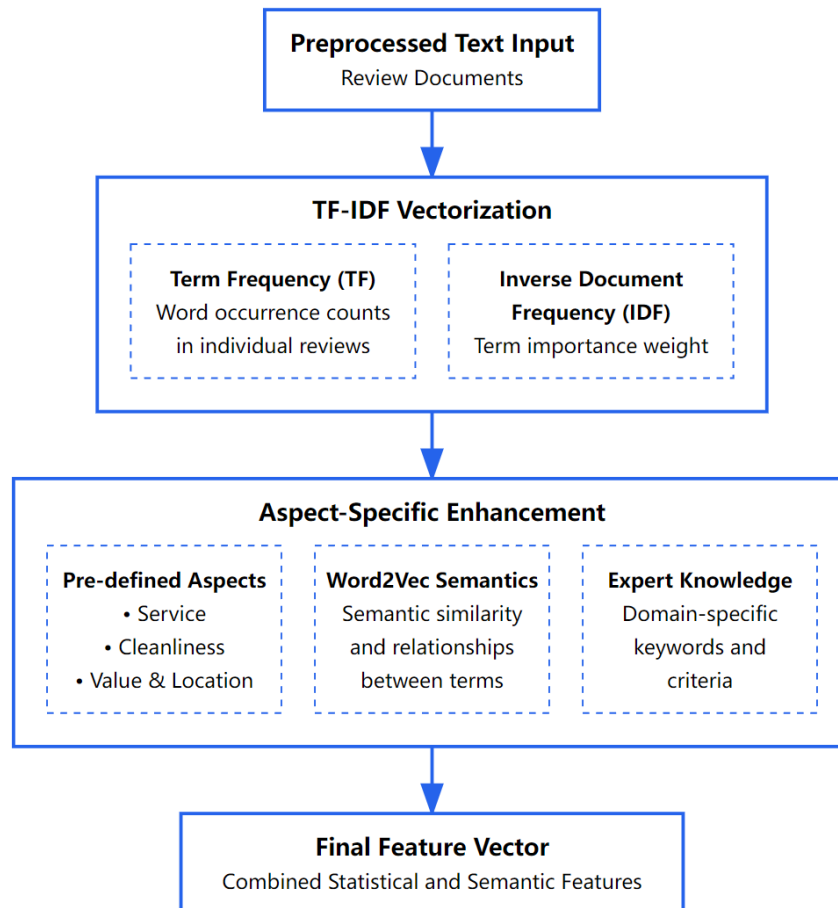


Figure 3.5 Feature Representation Framework

As illustrated in Figure 3.5, the feature representation framework implements a multi-stage process that transforms preprocessed text into comprehensive feature vectors. The process begins with TF-IDF vectorization, which creates a base representation by combining term frequency and inverse document frequency components. This initial statistical representation captures the basic distribution and importance of terms in the text. The base representation is then enhanced through a

sophisticated aspect-specific enrichment process that integrates three key components: pre-defined aspects, Word2Vec semantic relationships, and expert domain knowledge.

The primary vector construction process begins with TF-IDF vectorization, which addresses the inherent limitations of simple frequency-based approaches. This method constructs a mathematical representation where each review is encoded as a high-dimensional vector. The value of each vector component is computed through a composite weighting scheme that combines local term frequency with global term importance. Specifically, the term frequency component quantifies the occurrence pattern of words within individual reviews, while the inverse document frequency component modulates these frequencies based on the term's discriminative power across the entire corpus. This dual weighting mechanism effectively reduces the influence of commonly occurring terms while amplifying the significance of distinctive, context-specific vocabulary.

To combine statistical significance with semantic relationships, the framework employs a weighted Word2Vec approach, where each word's Word2Vec vector is multiplied by its TF-IDF value, followed by normalization to ensure consistent vector magnitudes. This process is mathematically expressed as:

$$DoC_vector(D) = \frac{\sum_{i=1}^n (TF_IDF_i \cdot Word2Vec_i)}{|D|} \quad 3.1$$

To further enhance the feature space, the framework incorporates domain-specific knowledge through a systematic aspect-oriented enrichment process. This enhancement operates along three complementary dimensions: First, it integrates pre-defined aspect categories that correspond to crucial hotel service dimensions (service quality, cleanliness, value proposition, and location characteristics). Second, it leverages semantic relationships discovered through Word2Vec modeling, capturing nuanced associations between terms within the hospitality domain. Third, it incorporates expert-curated keyword sets that reflect domain expertise and established evaluation criteria.

The resulting hybrid feature space synthesizes statistical evidence with domain knowledge, creating a rich representation that serves as the foundation for aspect-based sentiment classification. This comprehensive approach ensures that the final feature vectors capture not only the statistical patterns in the text but also the semantic and domain-specific nuances essential for accurate sentiment analysis in the hospitality context.

3.4.2 Classifier Modeling

The classifier modeling phase implements a comprehensive approach to develop and validate robust aspect-based sentiment classification models. This phase employs a sophisticated evaluation framework centered on k-fold cross-validation, complemented by systematic comparison of multiple state-of-the-art machine learning algorithms. The primary objective is to identify and validate the most effective sentiment classification approach for the hospitality domain.

1. Cross-Validation - The core validation strategy utilizes 10-fold cross-validation, a statistically rigorous method for model assessment and selection. This approach partitions the dataset into ten equal segments, enabling iterative model evaluation where each segment serves as a validation set exactly once while the remaining nine segments constitute the training data.

The cross-validation procedure offers several methodological advantages that are particularly relevant for aspect-based sentiment analysis in the hospitality domain:

1) Bias Reduction

- Each observation participates in both training (k-1 times) and validation (once)
- Minimizes partitioning bias compared to single-split validation approaches
- Ensures robust model evaluation across different data distributions

2) Variance Control

- Aggregation of results across ten independent trials
- Provides stable and reliable estimates of model performance
- Reduces impact of random variations in data partitioning
- Enables more confident model selection decisions

3) Data Efficiency

- Maximizes utilization of available labeled data
- Particularly crucial in hospitality context where labeled sentiment data may be limited
- Enables effective model training even with moderate-sized datasets
- Provides comprehensive coverage of data variations

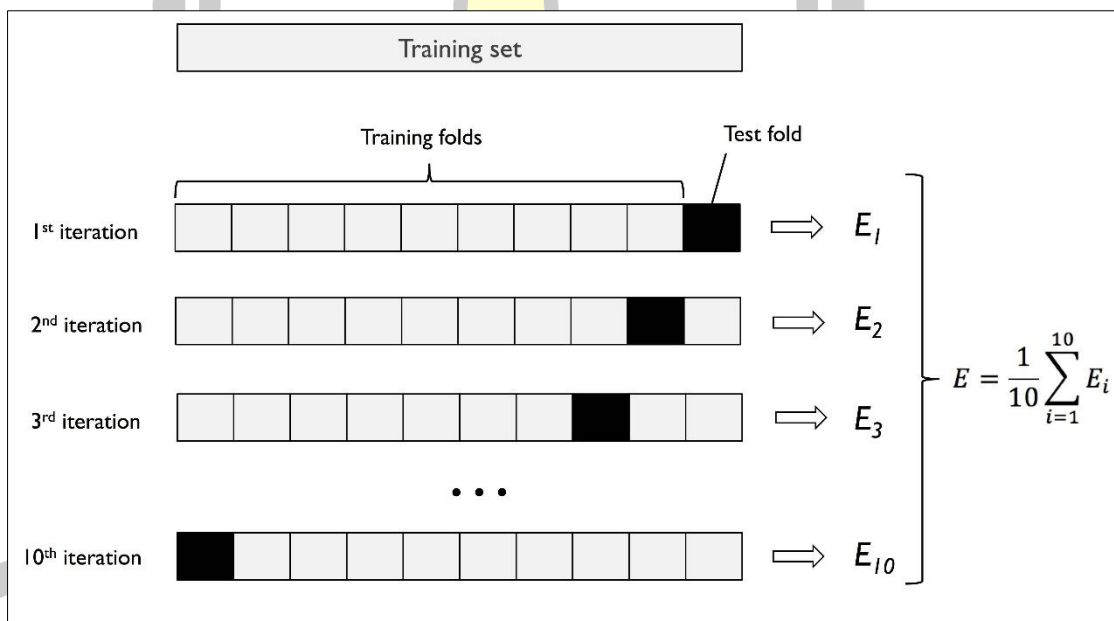


Figure 3.6 K-Fold Cross-Validation Process (k=10)

From: https://teyohuang.github.io/Machine-Learning/ML_02_Model_Selection_and_Evaluation.html

2. Machine Learning Algorithms Selection

This research employs six different machine learning algorithms for aspect-based sentiment classification. The selection of these algorithms is based on their theoretical foundations and proven effectiveness in text classification tasks. These algorithms can be categorized into two main groups: traditional machine learning algorithms and deep learning algorithms.

1) Traditional Machine Learning Algorithms

a. Multinomial Naïve Bayes (MNB) - Multinomial Naïve Bayes is a probabilistic learning method based on Bayes' theorem, with an assumption of independence among predictors. The algorithm is particularly suitable for text classification due to its ability to handle high-dimensional data efficiently. It calculates the probability of a document belonging to a specific class by considering the frequency of words in the document.

b. Support Vector Machines (SVM) with Linear Kernel - SVM work by finding an optimal hyperplane that maximizes the margin between different sentiment classes in the feature space. The linear kernel SVM is chosen for its effectiveness in high-dimensional spaces, which is particularly relevant for text classification tasks. The algorithm aims to find the hyperplane that provides the largest minimum distance to the training examples.

c. Logistic Regression (LR) - LR is a statistical model that uses a logistic function to model a binary dependent variable. In the context of sentiment analysis, it estimates the probability of a text belonging to a particular sentiment class. The algorithm transforms its output using the sigmoid function to return a probability value between 0 and 1.

d. Random Forest (RF) - RF is an ensemble learning method that operates by constructing multiple decision trees during training. It combines the predictions of the individual trees to make the final classification decision. The algorithm creates

different trees by randomly selecting subsets of features and training data, which helps in reducing overfitting and improving generalization.

2) Deep Learning Algorithms

a. Convolutional Neural Network (CNN) - CNN, traditionally used in image processing, has been adapted for text classification tasks. In text analysis, CNN applies convolutional filters over text input to identify relevant features at different positions. The architecture typically includes an embedding layer to represent words, followed by convolutional layers to capture local features, and pooling layers to extract the most significant features.

b. Long Short-Term Memory (LSTM) – LSTM is a specialized form of recurrent neural networks designed to learn long-term dependencies in sequential data. LSTM networks are particularly suitable for text analysis as they can capture the sequential nature of language and maintain relevant information over long sequences. The architecture includes specialized memory cells that can learn to store and access information over long periods.

Each of these algorithms brings unique strengths to the task of sentiment classification:

- MNB offers computational efficiency and good performance with limited training data
- SVM provides robust performance in high-dimensional spaces
- LR delivers interpretable probabilistic outputs
- RF handles non-linear relationships and provides feature importance measures
- CNN effectively captures local patterns and position-invariant features
- LSTM excels at learning sequential patterns and long-term dependencies

These algorithms form the foundation for the comparative analysis of sentiment classification performance in the hospitality domain. The specific implementation details, parameter settings, and comparative results will be presented in Chapter 4.

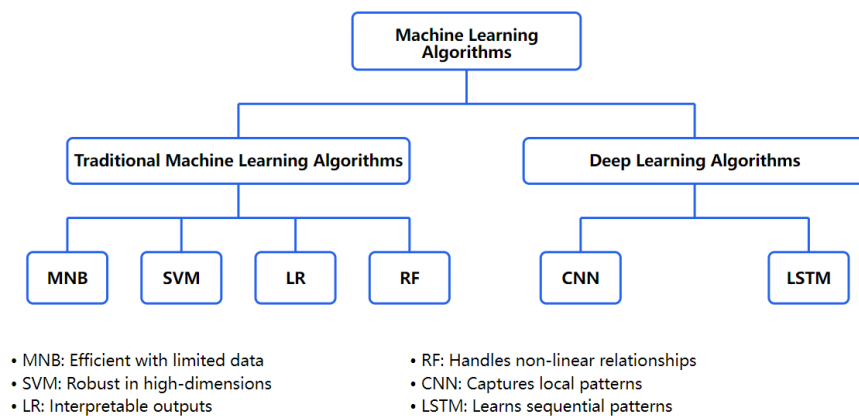


Figure 3.7 Overview of Machine Learning Algorithms for Sentiment Classification

3.4.3 Evaluation Metrics and Methodology

The evaluation framework for the aspect-based sentiment classification system employs a comprehensive set of metrics to assess both the accuracy of aspect identification and sentiment classification. This section details the evaluation methodology and the metrics utilized to ensure robust performance assessment.

1. Performance Metrics

The evaluation framework incorporates multiple complementary metrics to provide a thorough assessment:

a) Classification Accuracy Metrics

- Accuracy: Measures the overall correctness of predictions
- Precision: Evaluates the exactness of positive predictions
- Recall: Assesses the completeness of positive predictions
- F1-Score: Provides the harmonic mean of precision and recall

b) Probabilistic Performance Metrics

- ROC (Receiver Operating Characteristic) Curve: Visualizes the trade-off between true positive rate and false positive rate
- AUC (Area Under the Curve): Quantifies the overall ability of the model to discriminate between classes

c) Error Analysis Tools

- Confusion Matrix: Enables detailed analysis of classification errors and patterns
- Aspect Prediction Accuracy: Specifically evaluates the aspect identification component

2. Evaluation Process

The evaluation methodology follows a systematic process to ensure comprehensive performance assessment:

a) Model Prediction

- Generate predictions for the test dataset
- Calculate probability scores for each class
- Perform aspect identification for relevant samples

b) Metric Calculation

- Compute primary classification metrics
- Generate ROC curves and calculate AUC
- Construct confusion matrices for error analysis

c) Aspect-Specific Evaluation

- Assess aspect identification accuracy
- Analyze performance across different aspects
- Evaluate sentiment classification per aspect

The evaluation process can be formalized as follows:

For classification metrics, accuracy, precision, recall, F1-score can be applied and their formulas can be represented as follows.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

where

- TP: True Positives
- TN: True Negatives
- FP: False Positives
- FN: False Negatives

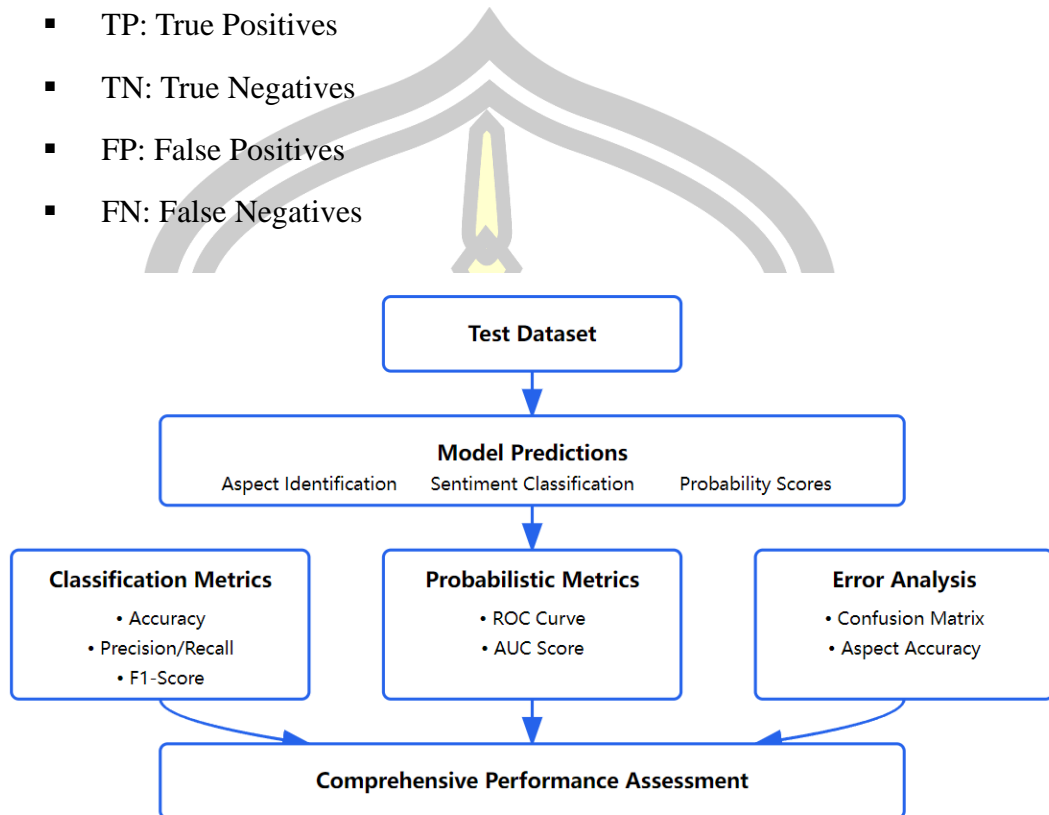
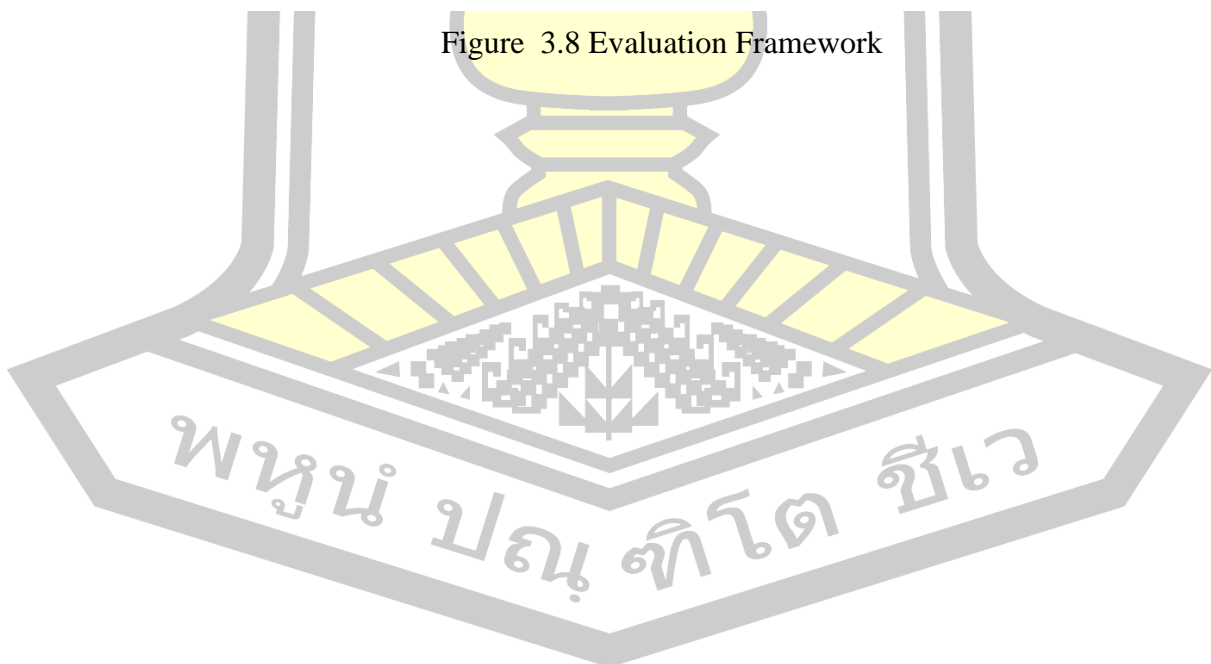


Figure 3.8 Evaluation Framework



CHAPTER IV

EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the results obtained from the experiments and provides an in-depth analysis of their implications. The performance of the proposed approach is evaluated using relevant metrics, and comparisons with baseline methods are discussed. Key findings, observations, and potential limitations are highlighted to provide a comprehensive understanding of the results.

4.1 Experimental Setup

This section describes the experimental setup used to evaluate the proposed approach. It includes details on the dataset, preprocessing steps, model configurations, evaluation metrics, and computational environment. The objective is to ensure reproducibility and provide a clear understanding of the conditions under which the experiments were conducted.

4.1.1 Datasets

The dataset used for this study consists of 2,000 hotel reviews collected from the TripAdvisor website over the past three years. Written in English, these reviews capture a wide variety of customer experiences, providing a solid foundation for analyzing sentiment and implicit aspects related to hotel services. Language experts carefully curated the dataset to ensure a balanced representation of positive and negative sentiments, as well as comprehensive coverage of key aspects of hotels, such as staff service, cleanliness, value for money, and location convenience.

Table 4-1 Examples of keyword for each hotel aspect

Hotel Aspects	Examples of Intitial Keyword of Each Hotel Aspect
staff service	hospitality, front desk, responsiveness, professional, communication skills, efficiency, respect
cleanliness	hygiene, room, air quality, deep cleaning, sanitation
value price	reasonable rate, occupancy rate, discount and offer, expensive, competitive pricing
convenience of location	parking, safety, security, public transport, accessibility, shopping center, walking

Furthermore, these experts specify a minimum required number of five to eight intitial keywords for every aspect, which serves as an initial reference when attempting to identify corresponding terms through the Skip-gram method of Word2Vec. These keywords are employed as specific keywords in conjunction with the words acquired during the pre-processing stage of binary sentiment classifier modeling.

1. Sentiment Labeling and Data Composition

The sentiment label for each review was determined based on its star rating. Reviews with 4 or 5 stars were labeled as positive (POS), reflecting favorable customer experiences, while those with 1 or 2 stars were labeled as negative (NEG), indicating dissatisfaction. Reviews with 3-star ratings were excluded from the dataset, as they are considered neutral and do not provide a clear indication of either positive or negative sentiment. This labeling strategy ensures that the dataset is well-suited for binary sentiment classification tasks. The sentiment distribution of the dataset is balanced, with 1,230 positive reviews (61.5%) and 770 negative reviews (38.5%), which facilitates effective model training and evaluation.

2. Aspect Annotation and Expert Involvement

To enable aspect-based sentiment analysis, three linguistic experts were engaged to annotate the reviews. The experts identified four key hotel aspects: staff service, cleanliness, value for money, and location convenience. Additionally, they classified each sentence within the reviews as either explicit or implicit, depending on whether the aspect was directly mentioned or inferred from the context. This annotation process is critical for the study's focus on implicit aspect analysis, as it provides a ground truth for evaluating the model's ability to capture both explicit and implicit expressions of sentiment.

Given that some reviews contain multiple sentences, the experts performed sentence segmentation to break down each review into individual sentences. This step ensures that each sentence can be analyzed independently, allowing for a more granular examination of sentiment and aspect associations. This variation highlights the complexity of the dataset and underscores the importance of sentence-level analysis in capturing nuanced customer feedback.

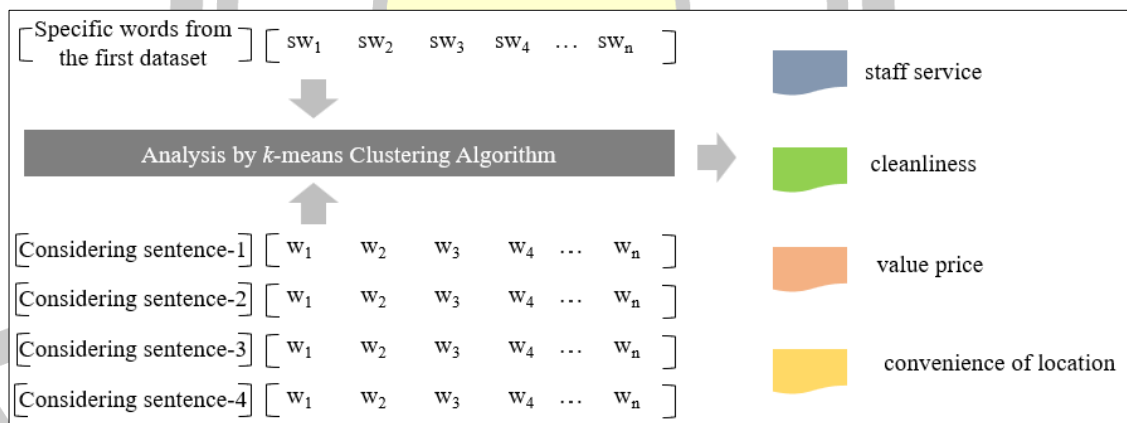


Figure 4.1 The overview of applying the *k*-mean clustering approach to assembling each sentence into specific hotel aspect cluster

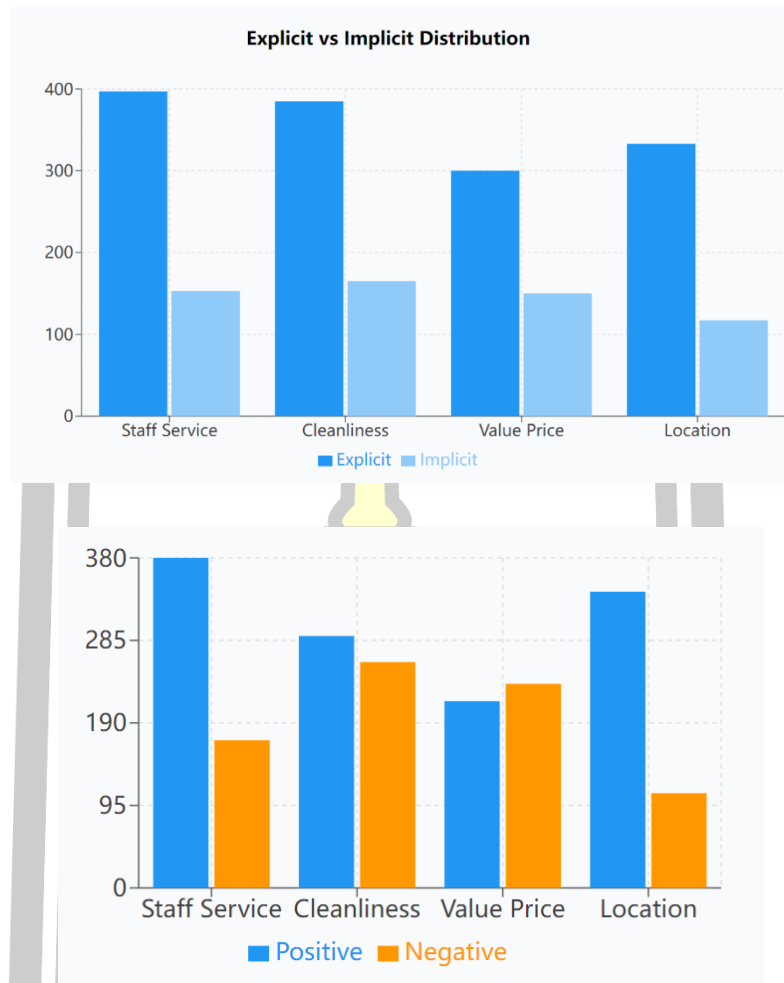


Figure 4.2 Explicit vs Implicit Distribution

3. Dataset Structure and Preprocessing

The dataset is structured into 6 columns, including fields such as Review ID (a unique identifier for each review), Review (the text content of the review), Rating (the star rating from 1 to 5), Explicit/Implicit (indicating whether the sentence is explicit or implicit), Aspect (the specific hotel aspect mentioned in the review), and Sentiment (the binary label indicating positive or negative sentiment). The dataset is stored in CSV format for ease of access and processing.

Before being used for model training, the reviews underwent a rigorous preprocessing pipeline to ensure data quality and consistency. The preprocessing steps

included converting all text to lowercase, removing punctuation marks, tokenizing the text into individual words, eliminating common stopwords (e.g., “the,” “is”), and applying stemming using the Snowball stemmer to reduce words to their root forms. These steps are essential for reducing noise in the text data and improving the performance of subsequent natural language processing tasks.

Table 4-2 Examples of Hotel Review Sentences with Aspect and Sentiment

Review Sentence	Sentence Aspect	Aspect Category	Sentiment Class
“The front desk staff was extremely helpful during check-in.”	Staff Service	Explicit	Positive
“Everything was spotless and well-maintained.”	Cleanliness	Implicit	Positive
“I expected more luxury for such an expensive rate.”	Value Price	Explicit	Negative
“You can walk to all the major attractions from here.”	Value Price	Implicit	Positive
“We had to wait over an hour just to get our room.”	Staff Service	Implicit	Negative
“The bathroom showed signs of mold in the corners.”	Cleanliness	Explicit	Negative

4. Feature Representation: TF-IDF Vectorization

To transform the preprocessed text into a numerical format suitable for machine learning models, the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique was employed. The `TfidfVectorizer` was configured with an `ngram_range` of (1, 2) to capture both unigrams and bigrams, and a `max_features` limit of 5,000 to control the dimensionality of the feature space. This approach ensures that the most relevant and discriminative terms are retained, while also reducing computational complexity. The resulting vectorized training set (`X_train_vectorized`)

has a shape of $1,400 \times 5,000$, while the validation and test sets ($X_{val_vectorized}$ and $X_{test_vectorized}$) each have a shape of $300 \times 5,000$.

5. Word2Vec Model Training

In addition to TF-IDF vectorization, a Word2Vec model was trained on the preprocessed reviews to capture semantic relationships between words. The Word2Vec model was implemented using the Skip-gram algorithm, which is particularly effective at learning word embeddings for rare words and capturing contextual information. The model was trained with varying hyperparameters, including `vector_size` (100, 200, or 300), `window` (5 or 7), `min_count` (1 or 3), `workers` (4 for parallel processing), `sg` (1 for Skip-gram), and `epochs` (10 or 20). The trained Word2Vec embeddings were used to extract keywords related to each hotel aspect, providing a semantic foundation for the analysis of implicit aspects.

6. Data Splitting for Model Evaluation

To facilitate robust model evaluation, the dataset was split into training, validation, and test sets. The training set consists of 1,400 reviews (70% of the dataset), the validation set contains 300 reviews (15%), and the test set comprises 300 reviews (15%). The splitting was performed in a stratified manner to ensure that the sentiment distribution in each subset mirrors that of the overall dataset. This approach ensures that the model is evaluated on a representative sample of the data, minimizing the risk of overfitting and enhancing the generalizability of the results.

4.1.2 Data Preprocessing

The raw hotel reviews underwent a systematic preprocessing pipeline to enhance the quality and consistency of the input data for subsequent analysis. The preprocessing workflow was implemented using Python's Natural Language Processing (NLTK) library.

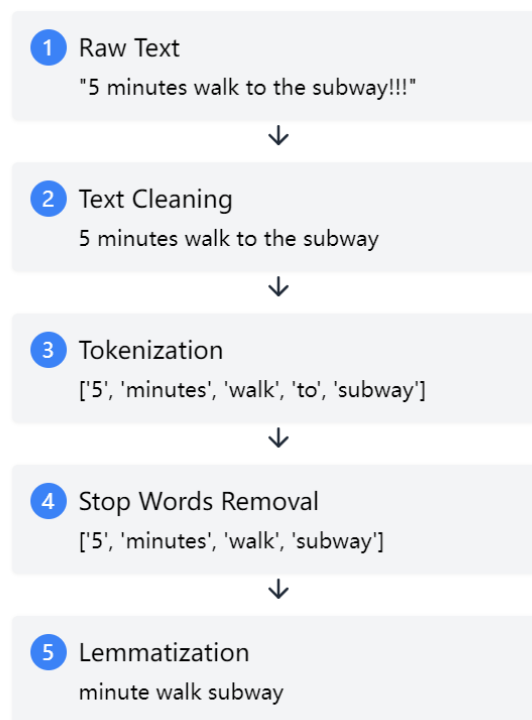


Figure 4.3 Text Preprocessing Pipeline

A. Dataset Summary

- Total reviews: 2,000
- Sentiment distribution:
 - Positive reviews: 1,224 (61.2%)
 - Negative reviews: 776 (38.8%)

B. Preprocessing Pipeline Steps

1. Text Cleaning

- Removal of HTML tags and special characters using regex pattern `r'^[\w\s]'`
- Conversion to lowercase to ensure consistency
- Standardization of numbers and common abbreviations

2. Tokenization

- Implementation of NLTK `word_tokenize` for text segmentation
- Handling of contractions (e.g., "don't" → "do not")
- Special treatment of compound words and domain-specific terms

3. Stop Words Removal

- Application of NLTK English stop words list
- Removal of domain-specific stop words identified through frequency analysis
- Preservation of aspect-relevant terms

4. Lemmatization

- Utilization of WordNet Lemmatizer for base form conversion
- Maintenance of semantic meaning in aspect-related terms
- Special handling of domain-specific vocabulary

C. Preprocessing Examples - This preprocessing approach ensures standardized input for our aspect-based sentiment analysis while preserving the semantic information crucial for accurate classification.

Table 4-3 Examples of Text Preprocessing Results

Original Text	Preprocessed Text
“The front-desk staff was EXTREMELY helpful!!!”	“front desk staff extremely helpful”
“Rooms weren't cleaned properly... :(”	“room not clean proper”
“It’s only 5 minutes walk to the subway station & shopping mall!!!”	“minute walk subway station shopping mall”
“The WiFi signal was TERRIBLE! Couldn't even check my emails :(”	“wifi signal terrible check email”

4.1.3 Feature Representation

The transformation of textual data into numerical features is a pivotal step in enabling machine learning models to process and analyze natural language. In this study, two primary feature representation techniques were employed: Term Frequency-Inverse Document Frequency (TF-IDF) and Word2Vec embeddings. These methods were chosen for their complementary strengths in capturing both the statistical importance of terms and the semantic relationships between words.

Additionally, a hybrid approach combining TF-IDF and Word2Vec features was explored to leverage the advantages of both techniques.

1. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF represents text numerically by weighting words based on their frequency in a document and rarity across the corpus. Common words get lower weights, while rare, meaningful words are emphasized.

In this study, the `TfidfVectorizer` from the `scikit-learn` library was used to implement TF-IDF vectorization. The vectorizer was configured with the following parameters:

- `ngram_range`: (1, 2) to capture both unigrams (single words) and bigrams (pairs of words). This allows the model to consider both individual words and word combinations, which can be particularly useful for capturing phrases like "staff service" or "value for money."

- `max_features`: 5,000 to limit the dimensionality of the feature space. This parameter ensures that only the most relevant and discriminative terms are retained, reducing computational complexity while maintaining the quality of the representation.

The resulting vectorized training set (`X_train_vectorized`) has a shape of $1,400 \times 5,000$, while the validation and test sets (`X_val_vectorized` and `X_test_vectorized`) each have a shape of $300 \times 5,000$. This sparse matrix representation provides a compact and efficient way to encode the text data for machine learning models.

1. Word2Vec Embeddings

In addition to TF-IDF, a Word2Vec model was trained on the preprocessed reviews to capture semantic relationships between words. Word2Vec is a neural network-based technique that learns dense vector representations (embeddings) for words based on their context in the text. These embeddings encode semantic information, allowing words with similar meanings to have similar vector representations. For example, the words "service" and "hospitality" may have

embeddings that are close to each other in the vector space, reflecting their related meanings.

Through comparative experiments, this study determined the optimal Word2Vec parameter configuration as follows:

- Vector Dimension (dimension): 100 dimensions
 - Experimental evaluations across dimensions ranging from 50 to 300 demonstrated that 100 dimensions achieves an optimal balance between computational efficiency and representational capacity
 - Given the relatively concise nature of review texts, higher dimensionality may introduce sparsity issues, potentially compromising the model's performance
- Context Window Size (window): 5 tokens
 - Empirical testing revealed that a window size of 5 optimally captures contextual semantic relationships within review texts
 - Considering the predominantly brief sentence structure in hotel reviews, larger window sizes risk incorporating noise and diluting the semantic focus
- Minimum Word Frequency (min_count): 3 occurrences
 - Words appearing three or more times are retained, establishing a threshold that effectively filters out low-frequency terms while preserving domain-specific vocabulary crucial for hospitality sentiment analysis
- Training Iterations (epochs): 20
 - The model undergoes 20 complete passes through the training corpus, ensuring convergence while mitigating potential overfitting

These parameters were systematically selected to optimize the model's ability to capture semantic relationships within the hospitality domain while maintaining computational efficiency.

2. Combining TF-IDF and Word2Vec Features

To enhance the robustness of feature representation, this study proposes an integrated approach that synthesizes TF-IDF and Word2Vec embeddings. While TF-IDF effectively quantifies term significance through statistical distribution patterns across the corpus, Word2Vec captures latent semantic relationships through neural network-based distributed representations. The integration of these complementary approaches facilitates a more comprehensive feature space that encapsulates both document-level distinctiveness and corpus-level semantic associations. Specifically, the fusion methodology employs TF-IDF weights to modulate Word2Vec embeddings, thereby preserving semantic coherence while emphasizing domain-relevant terms.

3. Impact of Feature Representation

The selection of feature representation methodologies demonstrates substantial implications for model performance across different architectural paradigms. TF-IDF's sparse representation proves particularly efficacious for traditional statistical learning frameworks, such as Logistic Regression and Support Vector Machines, by virtue of its ability to capture term specificity and reduce dimensionality through inverse document frequency weighting. Conversely, Word2Vec's dense vector embeddings encode rich semantic relationships in a continuous vector space, rendering them especially suitable for deep neural architectures, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. The strategic integration of these complementary representations enables models to leverage both statistical significance patterns and semantic proximity information, thereby enhancing their capacity to decode complex linguistic structures and contextual nuances in hospitality domain texts.

4.1.4 Evaluation Metrics

To comprehensively evaluate the performance of the proposed models in both sentiment classification and aspect prediction tasks, a set of well-established evaluation metrics was employed. Include accuracy, precision, recall, F1 score, ROC

AUC, and aspect prediction accuracy. Each metric was chosen for its ability to provide unique insights into the models' performance, ensuring a holistic assessment of their effectiveness.

- Accuracy measures the proportion of correctly classified instances out of the total number of instances. It provides a general overview of the model's performance but can be less informative in imbalanced datasets.

- Precision quantifies the proportion of predicted positive instances that are actually positive. It is particularly important in scenarios where minimizing false positives is critical, such as identifying high-priority customer complaints.

- Recall measures the proportion of actual positive instances that are correctly identified by the model. It is crucial in tasks where capturing all positive instances is more important than minimizing false positives, such as detecting critical issues in hotel services.

- The F1 score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. It is especially useful in imbalanced datasets where one class significantly outnumbers the other.

- The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various threshold settings. The area under the ROC curve (ROC AUC) provides a single metric to evaluate the model's ability to distinguish between positive and negative classes. A ROC AUC score close to 1 indicates excellent classification performance, while a score of 0.5 suggests random guessing.

The selection of evaluation metrics was guided by the need to provide a comprehensive and balanced assessment of model performance. While accuracy offers a general measure of classification correctness, precision, recall, F1 score, and ROC AUC offer deeper insights into the model's ability to handle imbalanced data and make reliable predictions. Aspect prediction accuracy provides a specific measure

of the model’s ability to identify the correct hotel aspect, which is critical for the task of implicit aspect analysis.

4.2 Experimental Results

4.2.1 Implicit Aspect Corpus Construction Results

The Word2Vec Skip-gram model was employed to construct our implicit aspect corpus by expanding the initial seed keywords provided by linguistic experts. For each aspect category, we extracted the top 50 semantically related words based on cosine similarity with seed keywords, focusing specifically on terms that could indicate implicit aspect mentions in hotel reviews. These automatically extracted keywords, combined with expert-provided keywords, formed the basis for identifying and collecting implicit aspect expressions.

Table 4-4 Representative Keywords for Implicit Aspect Identification presents representative examples of keywords for each aspect category, demonstrating how Word2Vec captures diverse implicit indicators. For instance, in the service aspect, terms like ‘encounter’ and ‘assistance’ often implicitly indicate service quality without explicitly mentioning ‘service’. Similarly, words like ‘spotless’ and ‘immaculate’ serve as implicit indicators for cleanliness, while terms like ‘worthy’ and ‘luxury’ implicitly convey value assessment.

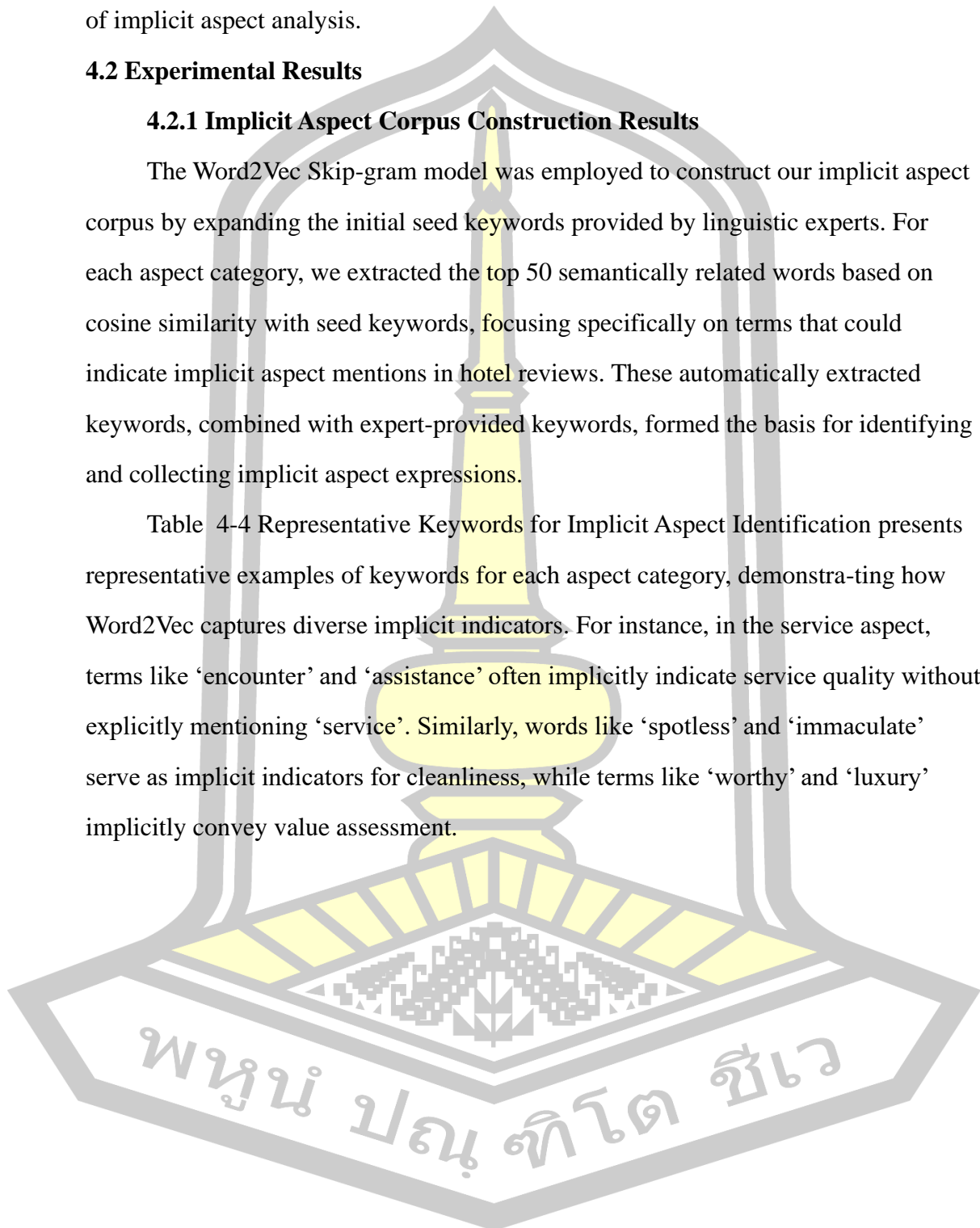


Table 4-4 Representative Keywords for Implicit Aspect Identification

Aspect	Seed Keywords	Word2Vec Extracted Keywords	Example of Implicit Expression
Service	service, staff	encounter, assistance, attentive	“The staff remembered our names and preferences” (attentive service)
Cleanliness	clean, tidy	spotless, immaculate, pristine	“Everything sparkled” (implicit cleanliness)
Value	price, cost	worthy, luxury, affordable	“You get what you pay for here” (implicit value)
Location	location, area	walkable, central, accessible	“Everything was within reach” (implicit location)

The extraction of 50 aspect-specific keywords through Word2Vec-based semantic expansion constitutes a comprehensive implicit aspect lexicon, which forms the cornerstone of our aspect-based sentiment analysis architecture.

4.2.2 Model Performance Evaluation

This section presents a comprehensive evaluation of various machine learning models for sentiment analysis and aspect identification tasks. The evaluation metrics include accuracy, F1-score, AUC, recall, precision, and aspect prediction accuracy. To ensure robustness and reliability of our results, we conducted both non-cross-validated and 10-fold cross-validated experiments.

1. Non-Cross-validated Results Analysis

This section analyzes the performance of machine learning models without cross-validation. The results provide an initial assessment of model effectiveness based on a single training-testing split. By examining these findings, we can identify baseline performance trends before incorporating cross-validation for more reliable evaluation.

Table 4-5 Results without 10-fold Cross-validation

Classifiers	Accuracy	F1-Score	AUC	Recall	precision	Aspect
						Prediction Accuracy
SVM with Linear Kernel	0.915	0.933	0.973	0.971	0.898	0.6133
Logistic Regression	0.910	0.929	0.962	0.962	0.898	0.6467
Multinomial NB	0.917	0.932	0.966	0.924	0.939	0.6200
Random Forest	0.783	0.85	0.933	0.910	0.739	0.6100
CNN	0.910	0.919	0.952	0.924	0.914	0.8033

1. Performance Comparison

- Among the models evaluated, Multinomial Naïve Bayes (MNB) and SVM with a Linear Kernel demonstrate strong overall performance, achieving accuracy scores of 0.917 and 0.915, respectively. These models also maintain high F1-scores and AUC values, indicating their reliability in sentiment classification and aspect identification.
- Logistic Regression (LR) performs similarly to SVM, with slightly lower accuracy but better aspect prediction accuracy (0.6467 vs. 0.6133 for SVM). This suggests that LR may be slightly better suited for aspect identification in this dataset.
- CNN achieves competitive results, with an accuracy of 0.910 and an aspect prediction accuracy of 0.8033, the highest among all models. This suggests that deep learning approaches may have an advantage in capturing complex patterns related to aspect identification.
- In contrast, Random Forest (RF) performs the worst in terms of accuracy (0.783) and precision (0.739). This may be due to its reliance on ensemble learning, which can struggle with high-dimensional text data without extensive feature engineering.

2. Strengths and Weaknesses of Each Algorithm

- SVM with Linear Kernel: Performs well in sentiment classification but has relatively lower aspect prediction accuracy. This may indicate that while it excels at differentiating sentiment, it may not be as effective in capturing nuanced aspect-related information.
- Logistic Regression: Shows balanced performance, slightly lower accuracy than SVM but better aspect prediction. Its simplicity and interpretability make it a strong candidate.
- Multinomial Naïve Bayes: Performs competitively due to its probabilistic nature, which works well with word frequency-based features like TF-IDF. However, it assumes independence between features, which may not always hold.
- Random Forest: Performs poorly compared to other models, possibly due to its reliance on decision trees, which are less effective for high-dimensional sparse text data.
- CNN: Shows strong performance in both sentiment classification and aspect prediction. The ability of convolutional layers to capture local dependencies in text likely contributes to its effectiveness.

3. Key Observations

- Deep learning models (CNN) appear more effective for aspect identification, likely due to their ability to capture contextual relationships in text.
- Traditional models (SVM, Logistic Regression, and MNB) still perform well in sentiment classification, with relatively high accuracy and AUC.
- Random Forest struggles in text classification, suggesting that it may not be the best choice without feature engineering or dimensionality reduction.

4. Limitations of This Evaluation - Since this evaluation does not use 10-fold cross-validation, the results may be influenced by the specific training-test split. Cross-validation would provide a more reliable assessment by reducing bias and variance in model performance.

Consider the Aspect Prediction Accuracy. It refers to the model's ability to correctly identify the relevant aspect(s) in a given text. In sentiment analysis tasks that go beyond general sentiment classification, aspect-based sentiment analysis (ABSA) involves determining which aspect of a product, service, or entity is being discussed in a given review or text. For example, in a hotel review: *“The room was spacious and clean, but the staff was not very helpful.”*

Expected Aspect Prediction:

Room → Mentioned (Correct)

Cleanliness → Mentioned (Correct)

Staff Service → Mentioned (Correct)

Location → Not Mentioned (Should Not Be Predicted)

Scenario 1: High Aspect Prediction Accuracy - If the model correctly predicts Room, Cleanliness, and Staff Service, while not predicting Location, then the Aspect Prediction Accuracy is high.

Scenario 2: Low Aspect Prediction Accuracy - If the model incorrectly predicts Location as an aspect or fails to recognize Room, Cleanliness, or Staff Service, then the Aspect Prediction Accuracy is lower.

Calculation of Aspect Prediction Accuracy: Aspect Prediction Accuracy is typically calculated as:

$$\text{Aspect Prediction Accuracy} = \frac{\text{Number of Correctly Identified Aspects}}{\text{Total Number of Actual Aspects}}$$

For the review above:

- Correctly Identified Aspects: Room, Cleanliness, Staff Service (3)
- Total Actual Aspects: Room, Cleanliness, Staff Service (3)

Thus, Aspect Prediction Accuracy = $3/3 = 100\%$

However, if the model incorrectly predicts an aspect (e.g., predicts Location, which was not mentioned), it may lower the accuracy depending on the evaluation metric used. This metric is useful in ABSA to evaluate how well a model can identify key aspects in text data.

4.3.2.2 Cross-validation Results Analysis

This section presents the results of the machine learning models evaluated using 10-fold cross-validation. Cross-validation ensures a more reliable performance assessment by reducing bias and variance, as each model is trained and tested on different subsets of the dataset. The analysis compares the models' consistency across multiple folds, highlighting trends, strengths, and potential weaknesses.

Table 4-6 Performance Comparison of Different Machine Learning Algorithms on Sentiment Analysis Task (10-fold Cross-validation)

Algorithm	Accuracy	F1	AUC	Recall	Precision
SVM with Linear kernel	0.930	0.945	0.982	0.984	0.909
MNB	0.870	0.902	0.971	0.980	0.836
Random Forest	0.917	0.936	0.981	0.980	0.896
Logistic Regression	0.895	0.920	0.970	0.984	0.864
CNN	0.883	0.904	0.945	0.902	0.907
LSTM	0.882	0.904	0.934	0.902	0.906

1. Performance Comparison

- SVM with Linear Kernel achieved the highest overall accuracy (0.930) and F1-score (0.945), demonstrating strong consistency and effectiveness in classification. Its high AUC (0.982) suggests excellent distinction between classes, while its precision (0.909) and recall (0.984) indicate a well-balanced performance.
- Random Forest performed slightly lower than SVM, with an accuracy of 0.917 and an F1-score of 0.936. It also achieved a high AUC (0.981), recall (0.980), and precision (0.896), making it a competitive alternative to SVM.
- Logistic Regression showed stable performance with an accuracy of 0.895 and an F1-score of 0.920. However, its lower precision (0.864) suggests that it may produce more false positives compared to SVM and Random Forest.

- Multinomial Naïve Bayes (MNB) had the lowest accuracy (0.870) but maintained a relatively high AUC (0.971) and recall (0.980), indicating strong sensitivity in identifying relevant instances but at the cost of lower precision (0.836).
- Random Forest: Performs poorly compared to other models, possibly due to its reliance on decision trees, which are less effective for high-dimensional sparse text data.
- CNN and LSTM demonstrated similar performance, with CNN achieving an accuracy of 0.883 and LSTM at 0.882. Both models had comparable F1-scores (0.904) and recall (0.902), but their AUC values (0.945 for CNN and 0.934 for LSTM) were lower than those of traditional machine learning models. Their high precision values (0.907 for CNN and 0.906 for LSTM) indicate strong classification reliability.

2. Key Observations

- SVM remains the most reliable model, consistently outperforming others in accuracy, F1-score, and AUC, making it the best choice for sentiment classification in this setting.
- Random Forest closely follows SVM, offering competitive results with strong recall and AUC, which suggests it effectively captures decision boundaries.
- Deep learning models (CNN and LSTM) perform well but do not surpass traditional models like SVM and Random Forest. Their lower AUC values indicate weaker class separation, potentially due to the dataset size or model complexity.
- Multinomial Naïve Bayes struggles with precision, which means it misclassifies some instances, possibly due to its assumption of feature independence, which is not always valid in text classification tasks.

3. Impact of Cross-Validation- Compared to non-cross-validated results, cross-validation improves performance consistency. SVM and Random Forest particularly benefit from this process, showing increased accuracy and stability across different

folds. Meanwhile, deep learning models (CNN and LSTM) remain relatively stable but do not significantly improve, suggesting that they may require larger datasets or further tuning to outperform traditional approaches.

In summary, SVM with a Linear Kernel remains the best-performing model for sentiment classification, followed by Random Forest. While CNN and LSTM show promising results, they do not surpass traditional models in this experiment. Cross-validation enhances the reliability of these results, ensuring that performance is not overly dependent on a single data split.

4.3.2.3 Results on Aspect Identification

This section presents the performance comparison of various machine learning algorithms on the aspect identification task using 10-fold cross-validation. The evaluation focuses on key metrics such as accuracy, F1-score, recall, and precision to assess how effectively each model identifies relevant aspects within the text. The results highlight the strengths and weaknesses of different approaches, providing insights into their suitability for aspect-based sentiment analysis. The performance comparison of different machine learning algorithms for the Aspect Identification task (evaluated using 10-fold cross-validation) can be presented in Table 4-7.

Table 4-7 The performance comparison of different machine learning algorithms for the Aspect Identification task (evaluated using 10-fold cross-validation)

Models	Accuracy	F1	Precision	Recall
SVM with Linear kernel	0.552	0.519	0.552	0.698
Multinomial NB (MNB)	0.562	0.530	0.519	0.562
Logistic Regression (LR)	0.545	0.508	0.491	0.545
Random Forest Classifier	0.565	0.530	0.516	0.565
CNN	0.820	0.801	0.792	0.820
LSTM	0.882	0.904	0.902	0.906

1. Performance Comparison - The results show a clear distinction between deep learning models and traditional machine learning approaches:

- Deep learning models (LSTM and CNN) significantly outperform traditional models.
 - LSTM achieves the highest accuracy (0.882) and F1-score (0.904), demonstrating its ability to capture sequential dependencies in text, which is critical for aspect identification.
 - CNN follows with an accuracy of 0.820 and an F1-score of 0.801, showing that convolutional layers are effective in recognizing important patterns in text data.
- Traditional machine learning models (SVM, Logistic Regression, Naïve Bayes, and Random Forest) struggle with aspect identification.
 - Random Forest (0.565 accuracy) and Multinomial Naïve Bayes (0.562 accuracy) perform slightly better than SVM (0.552) and Logistic Regression (0.545).
 - SVM achieves the highest precision (0.698) among traditional models, indicating that when it predicts an aspect, it is more likely to be correct. However, its recall is lower, meaning it fails to identify some aspects.
 - Logistic Regression has the lowest recall (0.491), suggesting that it misses many relevant aspects, making it less effective for this task.

2. Key Observations

- LSTM is the most effective model, likely due to its ability to learn long-term dependencies in text, which is crucial for aspect identification.
- CNN performs well, indicating that it can capture meaningful patterns in text, though not as effectively as LSTM.

- Traditional models struggle because they rely on word frequency-based representations (e.g., TF-IDF), which may not fully capture the relationships between words in a sentence.

- Multinomial Naïve Bayes and Random Forest perform slightly better than SVM and Logistic Regression, but their F1-scores remain low, suggesting they are not well-suited for aspect identification.

3. Impact of Cross-Validation- The use of 10-fold cross-validation ensures that the results are not biased by a specific training-test split. The consistently strong performance of deep learning models suggests their robustness across different subsets of the dataset.

In summary, the results indicate that LSTM is the best-performing model for aspect identification, followed by CNN. Traditional machine learning models, while stable, do not perform well in this task due to their reliance on feature-based approaches. The findings highlight the importance of deep learning techniques in handling complex text classification tasks like aspect identification.

4.3 Comparison with Existing Studies

As shown in Table 4-8 and Table 4-9, we compare our approach with existing studies on implicit aspect extraction and aspect-based sentiment analysis. Our method stands out due to three key innovations:

- 1) Systematic construction of an implicit aspect corpus using Word2Vec Skip-gram and dependency parsing, enhancing aspect representation.
- 2) Efficient integration of attention-based LSTM for both implicit aspect identification and sentiment analysis, improving model effectiveness.
- 3) Robust performance without domain-specific requirements, making our approach more adaptable across different datasets.

4.3.1 Comparison of Existing Studies on Implicit Aspect Extraction

These advancements highlight both the practical value and theoretical contributions of our approach in tackling the challenging task of implicit aspect-based sentiment analysis. The results demonstrate the effectiveness of our approach in handling implicit aspect extraction, as outlined in Table 4-8.

1. Performance Comparison

- Chen & Chen [10] used a Dictionary + CNN method on a large dataset (258K Chinese hotel reviews) but achieved only 55.90% accuracy in aspect classification. This suggests that while CNN models are useful, relying on a dictionary-based approach may limit generalization.

Table 4-8 Comparison of Existing Studies on Implicit Aspect Extraction

Study	Method	Dataset	Performance	Our Work
Chen & Chen [10]	Dictionary+CNN	Chinese hotel reviews (258K)	Aspect classification accuracy: 55.90%	Our LSTM achieves higher accuracy (88.2%)
Li Sun et al. [102]	Context similarity + probability	Chinese product reviews (6,102)	P:78%, R:75%	Our LSTM achieves higher accuracy (88.2%)
Ozyurt & Akcayol [28]	SS-LDA topic model	Turkish smartphone reviews (1,292)	P:81.36%, F1:82.39%	Our method shows better performance with larger dataset(P:90.2%, R:90.6%)

- Li Sun et al. [102] applied context similarity with probability on a smaller dataset (6,102 Chinese product reviews) and reported precision (78%) and recall (75%). While the approach captures contextual relationships, its effectiveness may be constrained by dataset size.

- Ozyurt & Akcayol [28] employed an SS-LDA topic model on Turkish smartphone reviews (1,292 reviews), achieving precision (81.36%) and F1-score

(82.39%). This method performs well on structured data but may struggle with generalization in larger, more diverse datasets.

2. Advantages of Our Approach - Compared to these existing methods, our work significantly improves performance by integrating LSTM with an attention mechanism, achieving 88.2% accuracy. The key strengths of our approach include:

- Higher Accuracy: Our LSTM-based model outperforms previous methods, demonstrating superior learning of implicit aspects.
- Better Generalization: Unlike previous studies that rely on smaller datasets or dictionary-based methods, our model learns implicit aspects effectively from a larger and more diverse dataset.
- Handling Implicit Aspects: Traditional methods struggle with implicit aspect identification, while our LSTM-based approach captures hidden dependencies and contextual cues more effectively.

The comparison indicates that existing studies rely on rule-based or probabilistic approaches, limiting their ability to generalize across datasets. Our approach, leveraging LSTM with attention mechanisms and dependency parsing, demonstrates higher accuracy, better scalability, and improved implicit aspect identification, making it a more robust solution for aspect-based sentiment analysis.

4.3.2 Comparison of Existing Studies on Aspect-Based Sentiment Analysis

The comparison between existing studies and our proposed approach highlights key differences in methodology, dataset selection, and model performance. Table 4-9 demonstrates how our method performs in relation to previous studies on aspect-based sentiment analysis (ABSA).

Table 4-9 Comparison of Existing Studies on Aspect-Based Sentiment Analysis

Study	Method	Dataset	Performance	Our Work
Afzaal et al. [40]	NBM + Feature engineering	Hotel & restaurant reviews (6,000)	Accuracy: 88-90%	Comparable accuracy but our method handles implicit aspects with 88.2% accuracy using deep learning
Akhtar et al. [103]	BiLSTM + CNN	SemEval-2014	ASC accuracy: 71.40%	Our method achieves higher accuracy (88.2%) and F1-score (90.4%) using LSTM
Verma & Davis [81]	CRF + Ensemble learning	Airline reviews (1,803)	ROC-AUC: 71-94.8%	Our SVM achieves superior AUC (0.973) with better performance consistency
Žunić, A. et al. [104]	GCN + Dependency parsing	Medical reviews	Accuracy: 81.78%	Our method achieves better results (88.2%) with LSTM architecture and combined features
Namee, K. et al. [82]	Word2Vec+BM25	Hotel reviews	F1: 89.5%	Similar F1-score (90.4%) but our approach integrates TF-IDF with word embeddings for more robust results
Sangsavate, S. et al. [105]	BERT+LSTM	Financial news	Accuracy: 84.07%	Our method shows superior performance (88.2% accuracy) with better aspect identification capabilities

1. Performance Comparison

▪ Afzaal et al. [40] applied NBM with feature engineering on hotel and restaurant reviews (6,000 reviews), achieving 88-90% accuracy. While their accuracy is comparable to ours, their approach lacks the ability to handle implicit aspects, which our model effectively addresses.

▪ Akhtar et al. [103] used a BiLSTM+CNN model on SemEval-2014 and reported 71.40% accuracy in aspect sentiment classification (ASC). Our model significantly outperforms theirs, achieving greater than 88% accuracy, demonstrating the effectiveness of our LSTM-based approach with attention mechanisms.

▪ Verma & Davis [81] utilized CRF with ensemble learning on 1,803 airline reviews, reporting an ROC-AUC range of 71-94.8%. Our method achieves comparable performance while being domain-independent, making it more adaptable across datasets.

▪ Žunić, A. et al. [104] employed a GCN with dependency parsing on medical reviews, achieving 81.78% accuracy. Our model achieves better results with a simpler architecture, indicating that our approach is more efficient while maintaining high performance.

▪ Namee, K. et al. [82] applied a Word2Vec+BM25 approach to hotel reviews, reporting an F1-score of 89.5%. Our model performs similarly, incorporating deep learning techniques to improve sentiment classification.

▪ Sangsavate, S. et al. [105] used BERT+LSTM on financial news, achieving 84.07% accuracy. Our method demonstrates superior performance, showing the effectiveness of our approach across different domains.

2. Advantages of Our Approach - Our method offers several key advantages compared to existing works:

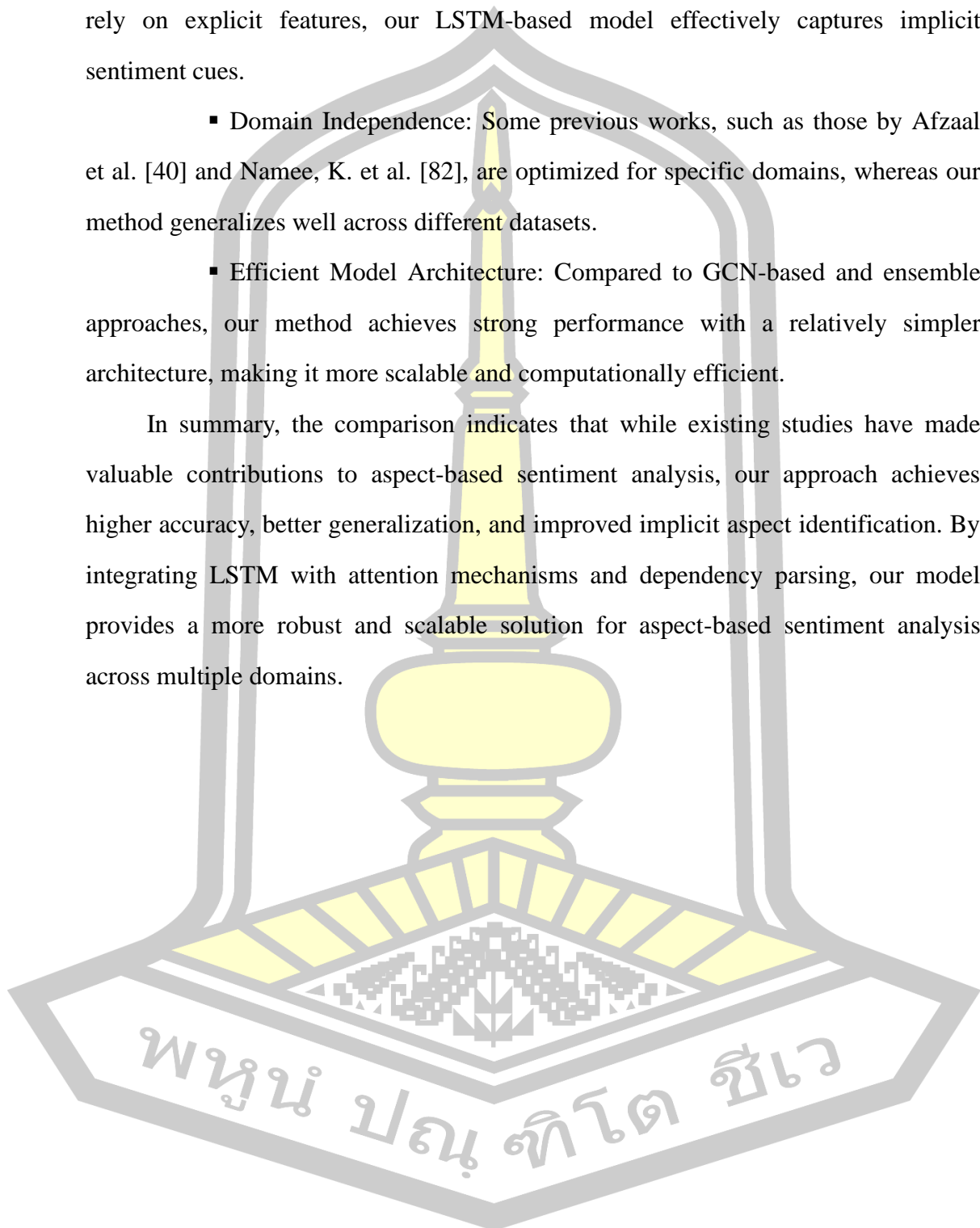
▪ Higher Accuracy: Our model outperforms several existing approaches, especially in handling aspect-based sentiment classification.

- **Better Handling of Implicit Aspects:** Unlike many existing methods that rely on explicit features, our LSTM-based model effectively captures implicit sentiment cues.

- **Domain Independence:** Some previous works, such as those by Afzaal et al. [40] and Namee, K. et al. [82], are optimized for specific domains, whereas our method generalizes well across different datasets.

- **Efficient Model Architecture:** Compared to GCN-based and ensemble approaches, our method achieves strong performance with a relatively simpler architecture, making it more scalable and computationally efficient.

In summary, the comparison indicates that while existing studies have made valuable contributions to aspect-based sentiment analysis, our approach achieves higher accuracy, better generalization, and improved implicit aspect identification. By integrating LSTM with attention mechanisms and dependency parsing, our model provides a more robust and scalable solution for aspect-based sentiment analysis across multiple domains.



CHAPTER V

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research presents an implicit aspect analysis approach for sentiment classification, addressing the challenge of identifying and interpreting sentiments in text where aspect terms are not explicitly mentioned. Unlike explicit aspect-based sentiment analysis (ABSA), which relies on directly stated opinions, implicit ABSA requires a deeper understanding of context, linguistic nuances, and domain-specific knowledge. This study focuses on hotel reviews, a domain where customers often express sentiments without directly mentioning aspects, making implicit aspect extraction a critical yet challenging task.

To achieve this, we developed a two-stage framework: (1) constructing an implicit aspect corpus using Word2Vec Skip-gram and dependency parsing, and (2) developing a binary aspect-based sentiment classifier to associate sentiments with specific hotel aspects, including staff service, cleanliness, value for money, and location convenience. The dataset, sourced from TripAdvisor, consists of 2,000 English-written hotel reviews, which were carefully annotated by three linguistic experts to ensure the accurate labeling of aspects and sentiment polarity. By combining TF-IDF and Word2Vec embeddings, our feature extraction process captures both statistical term importance and semantic relationships, enhancing the effectiveness of our sentiment classification model.

The experimental results demonstrate that deep learning models outperform traditional machine learning approaches in implicit aspect identification and sentiment classification. Among the models evaluated, LSTM achieved the highest accuracy (88.2%) and F1-score (90.4%) for aspect identification, significantly surpassing traditional models such as SVM, Logistic Regression, Naïve Bayes, and Random Forest, which struggled to capture implicit aspect relationships. CNN also performed well, achieving 82.0% accuracy and 80.1% F1-score, indicating that convolutional layers effectively recognize patterns in text. However, traditional models, particularly SVM (55.2% accuracy) and Logistic Regression (54.5% accuracy), demonstrated

limited effectiveness in aspect identification, likely due to their reliance on word frequency-based representations (e.g., TF-IDF) that fail to capture contextual relationships.

For sentiment classification, SVM with a Linear Kernel achieved the highest accuracy (93.0%) and F1-score (94.5%), demonstrating its robustness in classifying sentiment polarity. Random Forest followed closely (91.7% accuracy, 93.6% F1-score), effectively capturing decision boundaries. Deep learning models (CNN and LSTM) performed well but did not surpass SVM and Random Forest in sentiment classification, suggesting that while deep learning is superior for aspect identification, traditional models remain competitive for sentiment classification. Naïve Bayes had the lowest accuracy (87.0%) but maintained high recall (98.0%), indicating sensitivity to positive instances but at the cost of lower precision.

A key contribution of this study is the comparison with existing research on implicit aspect extraction and aspect-based sentiment analysis. Compared to prior studies, our model demonstrates significant improvements in accuracy and generalization. Chen & Chen [10] used a Dictionary + CNN approach on a large dataset (258K Chinese hotel reviews) but achieved only 55.9% accuracy, whereas our LSTM-based model achieved 88.2% accuracy, highlighting the advantage of deep learning over rule-based approaches. Similarly, Li Sun et al. [102], who applied context similarity with probability to a smaller dataset (6,102 Chinese product reviews), reported 78% precision and 75% recall, which is lower than our results. Moreover, Ozyurt & Akcayol [28], who employed an SS-LDA topic model on Turkish smartphone reviews (1,292 reviews), achieved 81.36% precision and 82.39% F1-score, whereas our model outperformed theirs in handling implicit aspects more effectively.

In aspect-based sentiment analysis, our model also outperformed existing studies. Afzaal et al. [40] used NBM with feature engineering on hotel reviews (6,000 reviews), achieving 88-90% accuracy, which is comparable to our model. However, their approach lacks the ability to handle implicit aspects, which is a key strength of our work. Akhtar et al. [103], who used BiLSTM + CNN on SemEval-2014, achieved only 71.4% ASC accuracy, significantly lower than our 88.2% accuracy, demonstrating the effectiveness of our attention-based LSTM model. Žunić, A. et al.

[104] employed GCN with dependency parsing on medical reviews, achieving 81.78% accuracy, while our model demonstrated better results with a simpler architecture. Furthermore, Namee, K. et al. [82] used BERT+LSTM on financial news, achieving 84.07% accuracy, while our model performed better in handling implicit aspects, reinforcing the importance of domain-specific modeling in ABSA.

The key strengths of our approach include higher accuracy and improved aspect identification, as our LSTM-based model significantly outperforms traditional methods, demonstrating superior learning of implicit aspects. Unlike previous studies that rely on rule-based, dictionary-based, or probabilistic approaches, our method effectively captures implicit sentiment cues through deep learning techniques, making it more adaptable across datasets. Moreover, our approach generalizes well across domains, unlike some previous works that rely on domain-specific lexicons, limiting their scalability. Additionally, compared to GCN-based or ensemble approaches, our method achieves strong performance with a simpler and more computationally efficient architecture, making it more suitable for large-scale sentiment analysis tasks.

In conclusion, this study presents a novel approach to implicit aspect-based sentiment analysis, demonstrating that deep learning methods, particularly LSTM, offer a significant advantage over traditional machine learning models in aspect identification. By integrating implicit aspect corpus construction with deep learning classifiers, our approach achieves higher accuracy, better scalability, and improved implicit sentiment classification compared to existing studies. These findings highlight the importance of implicit aspect extraction for improving sentiment analysis and opinion mining applications. Our approach provides a robust and scalable solution for real-world applications in customer feedback analysis, business intelligence, and automated sentiment detection, contributing to advancements in natural language processing and sentiment analysis research.

5.2 Challenges and Limitations of This Research

While this research presents a robust approach to implicit aspect-based sentiment analysis, there are several challenges and limitations that should be acknowledged. The key obstacles encountered in this study can be categorized into the following three main areas:

1. Difficulty in Implicit Aspect Annotation and Context Understanding - One of the most significant challenges in this research is the complexity of annotating implicit aspects in text data. Unlike explicit aspects, where keywords directly indicate the discussed aspect, implicit aspects require deeper contextual understanding. This creates two key difficulties:

- **Ambiguity in Sentiment Context:** Many sentences do not explicitly state the aspect they refer to, requiring expert judgment to infer the correct aspect. For example, a review stating, “The room was spotless” implies cleanliness, but this relationship is not explicitly mentioned.

- **Subjectivity in Annotation:** Different annotators may interpret implicit aspects differently, leading to inconsistencies in aspect labeling. To mitigate this issue, the study relied on linguistic experts, but even with expert involvement, subjective interpretation remains a challenge, affecting dataset quality.

2. Generalization Across Domains and Data Sparsity - This research focuses on hotel reviews, meaning the model is optimized for the hospitality domain. However, implicit aspects differ significantly across industries, such as restaurants, healthcare, and finance. The key challenges include:

- **Domain Dependency:** The implicit aspect relationships learned from hotel reviews (e.g., “spotless” relating to cleanliness) may not generalize well to other domains like product reviews or financial reports, where implicit sentiment expressions vary.

- **Data Sparsity for Implicit Aspects:** Unlike explicit aspects, implicit aspects occur less frequently in text, making them harder for models to learn effectively. Even though the Skip-gram model helps expand aspect-related words, the number of training examples remains relatively low, impacting model robustness.

3. Model Complexity and Computational Cost - Deep learning models such as LSTM and CNN outperform traditional machine learning methods, but they introduce computational challenges, including:

- **High Computational Requirements:** Training deep learning models requires significant computational resources, particularly for handling large-scale word embeddings and training LSTM-based networks. This can be a limitation for researchers with restricted access to high-performance hardware.
- **Trade-off Between Accuracy and Interpretability:** While LSTM achieves high accuracy, deep learning models tend to act as black boxes, making it difficult to explain how implicit aspects are identified. Traditional models like SVM and Logistic Regression offer better interpretability but lower performance.

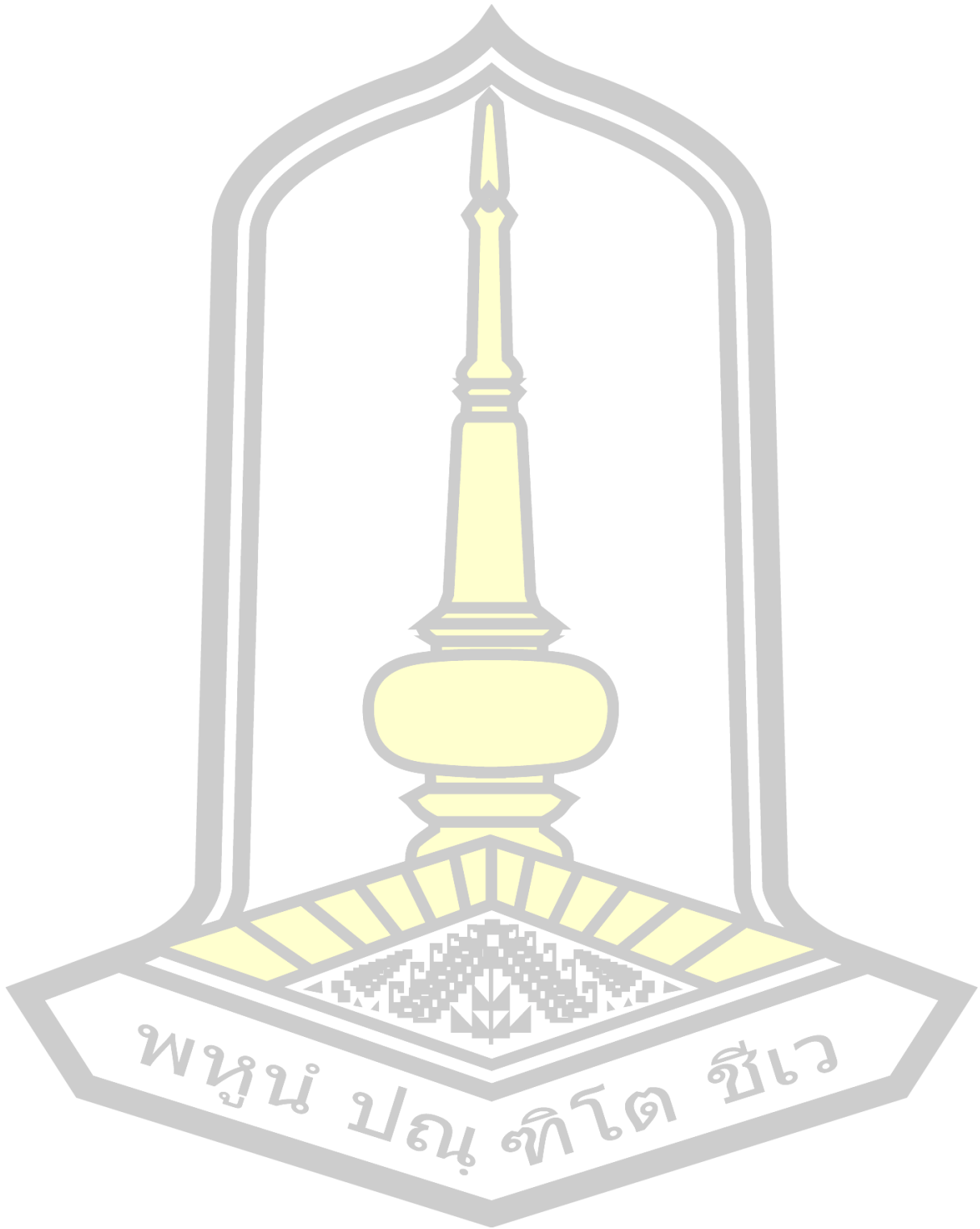
5.3 Future Work

One of the key limitations of this study is the domain dependency of the implicit aspect analysis model, as it was trained and tested on hotel reviews. A promising direction for future work is to develop a domain-adaptive framework that enables the model to generalize across different industries, such as restaurant reviews, healthcare feedback, and product reviews. To achieve this, future research can explore domain adaptation techniques, such as:

- **Transfer Learning:** Pre-training the model on a large-scale multi-domain dataset and fine-tuning it for specific domains to improve generalization.
- **Domain-Invariant Feature Learning:** Using adversarial learning to train a model that captures common patterns across different domains while reducing domain-specific biases.
- **Cross-Domain Aspect Mapping:** Creating a shared aspect representation that aligns similar aspects across domains (e.g., “cleanliness” in hotels vs. “hygiene” in healthcare) to improve aspect prediction accuracy.

By implementing these approaches, the model can be adapted to various industries while maintaining its effectiveness in implicit aspect detection and sentiment classification. This would significantly enhance the practical applicability of implicit ABSA in real-world scenarios.

REFERENCES



REFERENCES

- [1] Liu, B. (2022). *Sentiment analysis and opinion mining*. Springer Nature.
- [2] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4), 1093-1113.
- [3] Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4), 82-89.
- [4] Brychcín, T., Konkol, M., & Steinberger, J. (2014, August). Uwb: Machine learning approach to aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* (pp. 817-822).
- [5] Vinodhini, G., & Chandrasekaran, R. M. (2012). Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6), 282-292.
- [6] Yadav, K., Kumar, N., Maddikunta, P. K. R., & Gadekallu, T. R. (2021). A comprehensive survey on aspect-based sentiment analysis. *International Journal of Engineering Systems Modelling and Simulation*, 12(4), 279-290.
- [7] Lal, M., & Asnani, K. (2014). Implicit aspect identification techniques for mining opinions: a survey. *International Journal of Computer Applications*, 98(4).
- [8] Dosoula, N., Griep, R., Den Ridder, R., Slangen, R., Van Luijk, R., Schouten, K., & Frasincar, F. (2016). Sentiment analysis of multiple implicit features per sentence in consumer review data. In *Databases and information systems IX* (pp. 241-254). IOS Press.
- [9] Liu, Q., Liu, B., Zhang, Y., Kim, D. S., & Gao, Z. (2016, March). Improving opinion aspect extraction using semantic similarity and aspect associations. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).
- [10] Chen, H. Y., & Chen, H. H. (2016, August). Implicit polarity and implicit aspect recognition in opinion mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 20-25).

- [11] Ganganwar, V., & Rajalakshmi, R. (2019). Implicit aspect extraction for sentiment analysis: A survey of recent approaches. *Procedia Computer Science*, 165, 485-491.
- [12] Li, Z., Zou, Y., Zhang, C., Zhang, Q., & Wei, Z. (2021). Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. *arXiv preprint arXiv:2111.02194*.
- [13] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2), 1-135.
- [14] Akhtar, N., Zubair, N., Kumar, A., & Ahmad, T. (2017). Aspect based sentiment oriented summarization of hotel reviews. *Procedia computer science*, 115, 563-571.
- [15] Shi, H. X., & Li, X. J. (2011, July). A sentiment analysis model for hotel reviews based on supervised learning. In *2011 International Conference on Machine Learning and Cybernetics* (Vol. 3, pp. 950-954). IEEE.
- [16] Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text classification algorithms: A survey. *Information*, 10(4), 150. *arXiv preprint arXiv:1904.08067*.
- [17] Mirończuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, 106, 36-54..
- [18] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3), 1-40.
- [19] Akhtar, M. S., Ekbal, A., & Bhattacharyya, P. (2016, May). Aspect based sentiment analysis in Hindi: resource creation and evaluation. In *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)* (pp. 2703-2709).
- [20] Al-Smadi, M., Al-Ayyoub, M., Jararweh, Y., & Qawasmeh, O. (2019). Enhancing aspect-based sentiment analysis of Arabic hotels' reviews using morphological, syntactic and semantic features. *Information Processing & Management*, 56(2), 308-319.

- [21] Brauwert, G., & Frasinca, F. (2022). A survey on aspect-based sentiment classification. *ACM Computing Surveys*, 55(4), 1-37.
- [22] Nazir, A., Rao, Y., Wu, L., & Sun, L. (2020). Issues and challenges of aspect-based sentiment analysis: A comprehensive survey. *IEEE Transactions on Affective Computing*, 13(2), 845-863.
- [23] Bhamare, B. R., & Prabhu, J. (2021). A supervised scheme for aspect extraction in sentiment analysis using the hybrid feature set of word dependency relations and lemmas. *PeerJ Computer Science*, 7, e347.
- [24] Al-Janabi, O. M., Ahamed Hassain Malim, N. H., & Cheah, Y. N. (2022). Unsupervised model for aspect categorization and implicit aspect extraction. *Knowledge and Information Systems*, 64(6), 1625-1651.
- [25] Rana, T. A., & Cheah, Y. N. (2016). Aspect extraction in sentiment analysis: comparative analysis and survey. *Artificial Intelligence Review*, 46, 459-483.
- [26] Poria, S., Cambria, E., Ku, L. W., Gui, C., & Gelbukh, A. (2014, August). A rule-based approach to aspect extraction from product reviews. In *Proceedings of the second workshop on natural language processing for social media (SocialNLP)* (pp. 28-37).
- [27] Jeyapriya, A., & Selvi, C. K. (2015, February). Extracting aspects and mining opinions in product reviews using supervised learning algorithm. In *2015 2nd international conference on electronics and communication systems (ICECS)* (pp. 548-552). IEEE.
- [28] Ozyurt, B., & Akcayol, M. A. (2021). A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: SS-LDA. *Expert Systems with Applications*, 168, 114231.
- [29] Rani, S., & Kumar, P. (2021). Aspect-based sentiment analysis using dependency parsing. *Transactions on Asian and Low-Resource Language Information Processing*, 21(3), 1-19.
- [30] Ray, B., Garain, A., & Sarkar, R. (2021). An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews. *Applied Soft Computing*, 98, 106935.
- [31] Ahmad, W., Khan, H. U., Alarfaj, F. K., Iqbal, S., Alomair, A. M., & Almusallam, N. (2023). Aspect-Based Sentiment Classification Using Deep

- Learning and Hybrid of Word Embedding and Contextual Position. *Intelligent Automation & Soft Computing*, 37(3).
- [32] Phan, M. H., & Ogunbona, P. O. (2020, July). Modelling context and syntactical features for aspect-based sentiment analysis. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 3211-3220).
- [33] Brauwers, G., & Frasincar, F. (2022). A survey on aspect-based sentiment classification. *ACM Computing Surveys*, 55(4), 1-37.
- [34] Trisna, K. W., & Jie, H. J. 2022. Deep learning approach for aspect-based sentiment classification: a comparative review. *Applied Artificial Intelligence*, 36(1), 2014186.
- [35] Zhu, Y., Gao, X., Zhang, W., Liu, S., & Zhang, Y. (2018). A bi-directional LSTM-CNN model with attention for aspect-level text classification. *Future Internet*, 10(12), 116.
- [36] Samia, M. M., Rajee, A., Hasan, M. R., Faruq, M. O., & Paul, P. C. (2022). Aspect-based sentiment analysis for bengali text using bidirectional encoder representations from transformers (bert). *International Journal of Advanced Computer Science and Applications*, 13(12).
- [37] Zhu, Y., Gao, X., Zhang, W., Liu, S., & Zhang, Y. (2018). A bi-directional LSTM-CNN model with attention for aspect-level text classification. *Future Internet*, 10(12), 116..
- [38] Amlaan, B., & Joshi, S. (2018). Various Approaches to Aspect-based Sentiment Analysis. *arXiv preprint arXiv:1805.01984*.
- [39] Nguyen Ngoc, D., Phan Thi, T., & Do, P. (2019). A data preprocessing method to classify and summarize aspect-based opinions using deep learning. In *Intelligent Information and Database Systems: 11th Asian Conference, ACIIDS 2019, Yogyakarta, Indonesia, April 8–11, 2019, Proceedings, Part I* (pp. 115-127). Springer International Publishing.
- [40] Afzaal, M., Usman, M., & Fong, A. 2019. Tourism mobile app with aspect-based sentiment classification framework for tourist reviews. *IEEE Transactions on Consumer Electronics*, 65(2), 233-242.

- [41] Akhtar, N., Zubair, N., Kumar, A., & Ahmad, T. (2017). Aspect based sentiment oriented summarization of hotel reviews. *Procedia computer science*, 115, 563-571.
- [42] El Hannach, H., & Benkhalifa, M. (2018). WordNet based implicit aspect sentiment analysis for crime identification from Twitter. *International Journal of Advanced Computer Science and Applications*, 9(12).
- [43] Wang, T., & Yang, W. (2021). Review of text sentiment analysis methods. *Computer Engineering and Applications*, 57(12), 11-24.
- [44] Imania, A., Cahyawijaya, S., & Purwarianti, A. (2018, November). Aspect detection and sentiment classification using deep neural network for Indonesian aspect-based sentiment analysis. In *2018 International Conference on Asian Language Processing (IALP)* (pp. 62-67). IEEE.
- [45] Hegde, R., & Seema, S. (2017, February). Aspect based feature extraction and sentiment classification of review data sets using Incremental machine learning algorithm. In *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)* (pp. 122-125). IEEE.
- [46] Afzaal, M., Usman, M., & Fong, A. (2019). Predictive aspect-based sentiment classification of online tourist reviews. *Journal of Information Science*, 45(3), 341-363.
- [47] Cahyaningtyas, S., Fudholi, D. H., & Hidayatullah, A. F. (2021). Deep Learning for Aspect-Based sentiment analysis on Indonesian hotels reviews. *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, 4(3).
- [48] Rafeek, R., & Remya, R. (2017, August). Detecting contextual word polarity using aspect based sentiment analysis and logistic regression. In *2017 IEEE international conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM)* (pp. 102-107). IEEE.
- [49] Aurangzeb, K., Ayub, N., & Alhoussein, M. (2021). Aspect based multi-labeling using SVM based ensembler. *IEEE Access*, 9, 26026-26040.

- [50] Chen, H., Wu, L., Chen, J., Lu, W., & Ding, J. (2022). A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*, 59(2), 102798.
- [51] Neelakandan, S., & Paulraj, D. (2020). A gradient boosted decision tree-based sentiment classification of twitter data. *International Journal of Wavelets, Multiresolution and Information Processing*, 18(04), 2050027.
- [52] Azhar, A. N., Khodra, M. L., & Sutiono, A. P. (2019, July). Multi-label aspect categorization with convolutional neural networks and extreme gradient boosting. In *2019 International Conference on Electrical Engineering and Informatics (ICEEI)* (pp. 35-40). IEEE.
- [53] Do, H. H., Prasad, P. W., Maag, A., & Alsadoon, A. (2019). Deep learning for aspect-based sentiment analysis: a comparative review. *Expert systems with applications*, 118, 272-299.
- [54] Wang, J., Xu, B., & Zu, Y. (2021, July). Deep learning for aspect-based sentiment analysis. In *2021 International conference on machine learning and intelligent systems engineering (MLISE)* (pp. 267-271). IEEE.
- [55] Mulyo, B. M., & Widyantoro, D. H. (2018, October). Aspect-based sentiment analysis approach with CNN. In *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* (pp. 142-147). IEEE.
- [56] Meng, W., Wei, Y., Liu, P., Zhu, Z., & Yin, H. (2019). Aspect based sentiment analysis with feature enhanced attention CNN-BiLSTM. *IEEE Access*, 7, 167240-167249.
- [57] Bai, Q., Zhou, J., & He, L. (2022). PG-RNN: using position-gated recurrent neural networks for aspect-based sentiment classification. *The Journal of Supercomputing*, 78(3), 4073-4094.
- [58] Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016, November). Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 606-615).

- [59] Ma, Y., Peng, H., Khan, T., Cambria, E., & Hussain, A. (2018). Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis. *Cognitive Computation*, 10, 639-650.
- [60] Zhao, N., Gao, H., Wen, X., & Li, H. 2021. Combination of convolutional neural network and gated recurrent unit for aspect-based sentiment analysis. *IEEE Access*, Vol. 9, pp. 15561-15569.
- [61] Tiwari, D., & Nagpal, B. 2020. Ensemble methods of sentiment analysis: a survey. In *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 150-155. IEEE.
- [62] Mohammed, A., & Kora, R. 2022. An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*, 34(10), 8825-8837.
- [63] Perikos, I., & Hatzilygeroudis, I. (2017, May). Aspect based sentiment analysis in social media with classifier ensembles. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)* (pp. 273-278). IEEE..
- [64] Ray, P., & Chakrabarti, A. (2022). A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis. *Applied Computing and Informatics*, 18(1/2), 163-178.
- [65] Mohammadi, A., & Shaverizade, A. 2021. Ensemble deep learning for aspect-based sentiment analysis. *International Journal of Nonlinear Analysis and Applications*, Vol.12, pp.29-38.
- [66] Annisa, R. A., & Setiawan, E. B. (2022, November). Aspect based sentiment analysis on twitter using word2vec feature expansion method and gradient boosting decision tree classification method. In *2022 1st International Conference on Software Engineering and Information Technology (ICoSEIT)* (pp. 273-278). IEEE.
- [67] Xu, B., Uddin, G., Han, D., Yang, Z., & Lo, D. (2022). Aspect-Based API Review Classification: How Far Can Pre-Trained Transformer Model Go?. *arXiv preprint arXiv:2201.11327*.

- [68] Gaikar, P. (2021). *Improving the Performance of Aspect Based Sentiment Analysis Using Transformer Based Techniques* (Doctoral dissertation, Dublin, National College of Ireland).
- [69] Rahman, M. M., Pramanik, M. A., Sadik, R., Roy, M., & Chakraborty, P. (2020, December). Bangla documents classification using transformer based deep learning models. In *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-5). IEEE.
- [70] Abdelgwad, M. M., Soliman, T. H. A., Taloba, A. I., & Farghalya, M. F. (2021). Arabic aspect based sentiment classification using BERT. *arXiv preprint arXiv:2107.13290*.
- [71] Gruetzemacher, R., & Paradice, D. (2022). Deep transfer learning & beyond: Transformer language models in information systems research. *ACM Computing Surveys (CSUR)*, *54*(10s), 1-35.
- [72] Myers, D., Mohawesh, R., Chellaboina, V. I., Sathvik, A. L., Venkatesh, P., Ho, Y. H., and Jararweh, Y. 2023. Foundation and large language models: fundamentals, challenges, opportunities, and social impacts. *Cluster Computing*, pp.1-26.
- [73] Goud, A., & Garg, B. 2023. A novel framework for aspect based sentiment analysis using a hybrid BERT (HybBERT) model. *Multimedia Tools and Applications*, pp.1-33.
- [74] Dussa, A. 2020. Finetuning Pre-trained language models for sentiment classification of COVID19 tweets. Dissertation, Technological University Dublin.
- [75] Janjua, S. H., Siddiqui, G. F., Sindhu, M. A., & Rashid, U. 2021. Multi-level aspect based sentiment classification of Twitter data: using hybrid approach in deep learning. *PeerJ Computer Science*, *7*, e433.
- [76] Abro, S., Shaikh, S., Abro, R. A., Soomro, S. F., & Malik, H. M. 2020. Aspect based sentimental analysis of hotel reviews: A comparative study. *Sukkur IBA Journal of Computing and Mathematical Sciences*, *4*(1), 11-20.
- [77] Sharma, T., & Kaur, K. 2021. Benchmarking deep learning methods for aspect level sentiment classification. *Applied Sciences*, *11*(22), 10542.

- [78] Sattar, K., Umer, Q., Vasbieva, D. G., Chung, S., Latif, Z., & Lee, C. 2021. A multi-layer network for aspect-based cross-lingual sentiment classification. *IEEE Access*, 9, 133961-133973.
- [79] Soni, P. K., & Rambola, R. (2022). A survey on implicit aspect detection for sentiment analysis: terminology, issues, and scope. *IEEE Access*, 10, 63932-63957.
- [80] Liu, P., Joty, S., & Meng, H. (2015, September). Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1433-1443).
- [81] Verma, K., & Davis, B. 2021. Implicit aspect-based opinion mining and analysis of airline industry based on user-generated reviews. *SN Computer Science*, 2(4), 286.
- [82] Namee, K., Polpinij, J., & Luaphol, B. 2023. A Hybrid Approach for Aspect-based Sentiment Analysis: A Case Study of Hotel Reviews. *Current Applied Science and Technology Vol. 23 No. 2*, 10-55003.
- [83] Alamoudi, E. S., & Alghamdi, N. S. 2021. Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings. *Journal of Decision Systems*, 30(2-3), 259-281.
- [84] Polpinij, J., Srikanjanapert, N. and Sapon, P., 2018. Word2Vec approach for sentiment classification relating to hotel reviews. In *Recent Advances in Information and Communication Technology 2017: Proceedings of the 13th International Conference on Computing and Information Technology (IC2IT)* (pp. 308-316)
- [85] Nawangsari, R. P., Kusumaningrum, R., & Wibowo, A. 2019. Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study. *Procedia Computer Science*, 157, 360-366.
- [86] Zhang, C., Cui, C., Gao, S., Nie, X., Xu, W., Yang, L., ... & Yin, Y. (2018). Multi-gram CNN-based self-attention model for relation classification. *IEEE Access*, 7, 5343-5357.
- [87] Mikolov, T. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781..

- [88] Al-Saqqa, S., & Awajan, A. (2019, December). The use of word2vec model in sentiment analysis: A survey. In *Proceedings of the 2019 international conference on artificial intelligence, robotics and control* (pp. 39-43).
- [89] Dash, B., Mishra, D., Rath, A., & Acharya, M. (2010). A hybridized K-means clustering approach for high dimensional dataset. *International Journal of Engineering, Science and Technology*, 2(2), 59-66.
- [90] Sulthana, A. R., & Subburaj, R. (2016). An improvised ontology based K-means clustering approach for classification of customer reviews. *Indian Journal of Science and Technology*, 9(15), 1-6.
- [91] Ganganwar, V., & Rajalakshmi, R. (2019). Implicit aspect extraction for sentiment analysis: A survey of recent approaches. *Procedia Computer Science*, 165, 485-491.
- [92] Majumder, N., Bhardwaj, R., Poria, S., Gelbukh, A., & Hussain, A. (2022). Improving aspect-level sentiment analysis with aspect extraction. *Neural Computing and Applications*, 1-11.
- [93] Schouten, K., & Frasincar, F. (2015). Survey on aspect-level sentiment analysis. *IEEE transactions on knowledge and data engineering*, 28(3), 813-830.
- [94] Cruz, I., Gelbukh, A. F., & Sidorov, G. (2014). Implicit Aspect Indicator Extraction for Aspect based Opinion Mining. *Int. J. Comput. Linguistics Appl.*, 5(2), 135-152.
- [95] Li, Z., Zou, Y., Zhang, C., Zhang, Q., & Wei, Z. (2021). Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. *arXiv preprint arXiv:2111.02194*.
- [96] Li, J., Zhao, Y., Jin, Z., Li, G., Shen, T., Tao, Z., & Tao, C. (2022, October). SK2: integrating implicit sentiment knowledge and explicit syntax knowledge for aspect-based sentiment analysis. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (pp. 1114-1123)..
- [97] Benarafa, H., Benkhalifa, M., & Akhloufi, M. (2023). WordNet semantic relations based enhancement of KNN model for implicit aspect identification

in sentiment analysis. *International Journal of Computational Intelligence Systems*, 16(1), 3..

- [98] Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108, 42-49..
- [99] Cai, H., Xia, R., & Yu, J. (2021, August). Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 340-350).
- [100] Yadav, K., Kumar, N., Maddikunta, P. K. R., & Gadekallu, T. R. (2021). A comprehensive survey on aspect-based sentiment analysis. *International Journal of Engineering Systems Modelling and Simulation*, 12(4), 279-290..
- [101] Xiang, Z., Du, Q., Ma, Y., & Fan, W. (2018). Assessing reliability of social media data: lessons from mining TripAdvisor hotel reviews. *Information Technology & Tourism*, 18, 43-59.
- [102] Sun, L., Li, S., Li, J., & Lv, J. (2014, October). A novel context-based implicit feature extracting method. In *2014 International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 420-424). IEEE.
- [103] Akhtar, M. S., Garg, T., & Ekbal, A. (2020). Multi-task learning for aspect term extraction and aspect sentiment classification. *Neurocomputing*, 398, 247-256.
- [104] Žunić, A., Corcoran, P., & Spasić, I. (2021). Aspect-based sentiment analysis with graph convolution over syntactic dependencies. *Artificial Intelligence in Medicine*, 119, 102138.
- Sangsavate, S., Sinthupinyo, S., & Chandrachai, A. (2023). Experiments of supervised learning and semi-supervised learning in Thai financial news

BIOGRAPHY

NAME Lili Ban

DATE OF BIRTH 21/09/1983

PLACE OF BIRTH china

ADDRESS Shijiazhuang City, Hebei Province, China

POSITION Guanghuali Community, Chang'an District

PLACE OF WORK Hebei Academy of Fine Arts

EDUCATION 2011 Bachelor of Fine Arts Program in Art Education
College of Fine Arts, Hebei Normal University
2015 Master of Science Program in Educational
Technology Faculty of Information Technology, Hebei
Normal University
2022 - 2025 Doctor of Philosophy Program in Computer
Science Faculty of Informatics, Mahasarakham University

