VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

LE MINH BINH

IMPROVING ASPECT-ORIENTED SENTIMENT ANALYSIS FOR VIETNAMESE E-COMMERCE DATA USING SEMI-AUTOMATIC DICTIONARY FEATURES

GRADUATION THESIS

Major: Computer Science

HANOI - 2022

VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

LE MINH BINH

IMPROVING ASPECT-ORIENTED SENTIMENT ANALYSIS FOR VIETNAMESE E-COMMERCE DATA USING SEMI-AUTOMATIC DICTIONARY FEATURES

GRADUATION THESIS

Major: Computer Science

Supervisor: Assoc. Prof. Tran Trong Hieu

Co-Supervisor: MSc. Can Duy Cat

HANOI - 2022

Abstract

With e-commerce exploding in popularity in Vietnam, the volume of consumer feedback data on products on e-commerce platforms is constantly expanding. This situation presents a problem for business units seeking to comprehend customers through the use of this massive data stream, necessitating the development of automated systems capable of deciphering client opinions expressed through comments. ABSA is the problem of identifying sentiment elements of interest in a text, either as a single element or multiple elements with a dependency relationship. For a product on an e-commerce platform, the problem focuses on discovering product-related characteristics concealed in user reviews and ratings, as well as the speaker's positive/negative attitude about each aspect.

This thesis describes a method that was designed to deal with aspect-based sentiment classification for Vietnamese e-commerce reviews. Based on the approach of incorporating lexical priors into topic models of Jagarlamudi et al., the thesis proposes a model to solve the thesis problem focusing on the component which generate dictionary features using user-defined words by SeededLDA, an improved version of Latent Dirichlet Allocation, one of the most known topic modeling algorithms.

This approach achieved the maximum F1 of 99.3% and 90% in terms of "service" aspect for positive and negative label respectively, and 13.5% improvement in macro-f1 compared to baseline model. The results demonstrate that the method utilized in this thesis is effective.

***Keywords**: Sentiment Analysis, Aspect-Based Sentiment Analysis, Vietnamese multi-aspect dataset, Latent Dirichlet Allocation*

Acknowledgements

Allow me to begin by expressing my heartfelt gratitude to my supervisors, Assoc. Prof. Tran Trong Hieu and MSc. Can Duy Cat. The teachers' dedicated instruction and close monitoring from the time I first learned about the topic, progressing step by step until I completed this thesis, provided me with invaluable assistance in developing a new perspective on the subject, both theoretically and practically.

Additionally, I am deeply indebted to MSc. Le Hoang Quynh for her patience guidance from day one, her conscientious orientation, unwavering support, and motivational example in and out of class.

Furthermore, I would like to express my gratitude to the Data Science and Knowledge Technology Laboratory for providing me with the opportunity to study and conduct scientific research. I was able to complete this thesis in a systematic and scientific manner as a result of these experiences.

I would like to express my sincere thanks to my friend Le Thi Phuong for her kind and dedicated help since we started to participate in scientific research, especially in technical implementation.

Finally, I would like to express my gratitude to my family and friends for their support and encouragement throughout the process of writing this thesis.

Declaration

I declare that the thesis has been composed by myself and that the work has not be submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work which has formed part of jointly-authored publications has been included. My contribution and those of the other authors to this work have been explicitly indicated below. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I take full responsibility and take all prescribed disciplinary actions for our commitments.

I declare that this thesis has not been submitted for a higher degree to any other University or Institution.

Student

Le Minh Binh

Table of Contents

Acronyms

ABSA	Aspect-Based Sentiment Analysis
KNN	K-nearest Neighbors
LDA	Latent Dirichlet Allocation
LR	Logistics Regression
MWLA	Mini-Window Locating Attention
NB	Naive Bayes
OH	One-Hot encoding
RF	Random Forest
SA	Sentiment Analysis
SVM	Suport Vector Machine

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

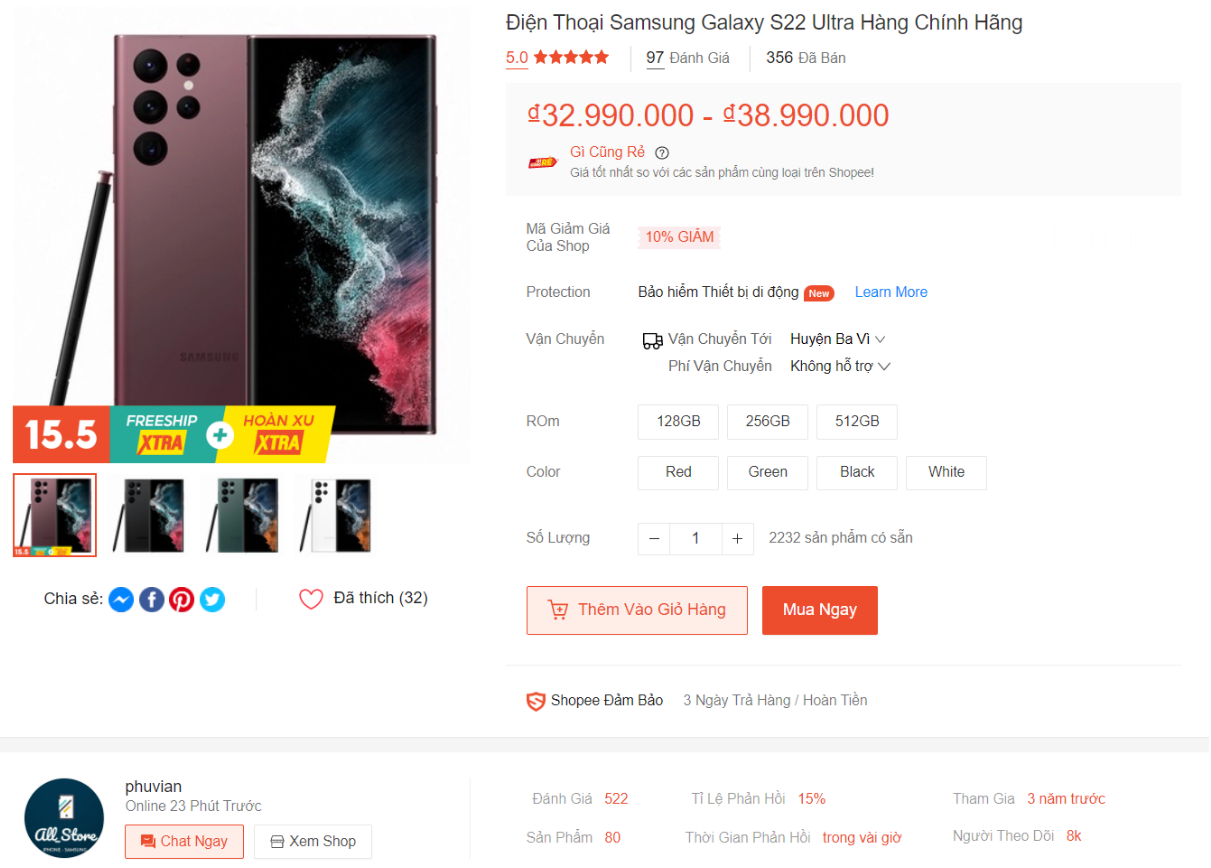
The last few years have seen an explosion in online shopping, and Vietnam is no exception to this trend. This is especially true in the context of the COVID-19 epidemic, which has had an impact on consumers' shopping habits, transforming e-commerce platforms into more familiar shopping channels for them. According to the Google e-Conomy 2021[1] research, the gross merchandise value of the Vietnamese e-commerce market climbed by 53 percent in 2020, reaching 13 billion USD. By 2025, this value will more than triple to 39 billion.

It is necessary to have a general understanding of the purchasing process in order to be able to identify the factors that influence a customer's purchasing decision process. According to Kotler[5], that process includes steps: need recognition, information search, evaluation of alternatives, purchase decision and post-purchase behavior. We can observe one element that appears throughout the majority of the purchasing process, and that is the research and evaluation of products. A customer searches for a product that they need to buy, interested in purchasing, or the one that the system has recommended. They will then look for figures like the number of purchases, the average number of star reviews for that product; or some other number of the store that is selling such as the number of reviews, response rate, number of followers, etc. A good example of a product on e-commerce platforms can be shown as Figure 1.1. Some of the information that can be observed to evaluate a seller include: "Đánh giá" (Rating), "Tỉ lệ phản hồi" (Response Rate), "Thời gian phản hồi" (Response Time), "Người theo dõi" (Followers)

[1] https://economysea.withgoogle.com/

.

However, the numbers do not appear to be sufficient. Also according to Kotler, one of the two factors that can come between a buyer's purchase intention and a purchase decision is the attitudes of others. Opinions are one of the most important factors influencing our every single behavior, as we are constantly interested in hearing what other people think about a topic before making a decision. Considered in the context of Vietnam, where tens of millions of people use E-commerce platforms in their daily lives, it is not difficult for one person to find hundreds, thousands, or even tens of thousands of people who share a common interest in a product that they are considering purchasing. Reading the reviews of these people about the product in which they are interested is a great way to determine whether or not to purchase the product being discussed. This type of feedback, however, can have a negative impact on the shopping experience when buyers are completely overwhelmed by a large number of diverse comments, both in terms of sentiment and the specific aspects mentioned.



Điện Thoại Samsung Galaxy S22 Ultra Hàng Chính Hãng

5.0 ★★★★★ | 97 Đánh Giá | 356 Đã Bán

₫32.990.000 - ₫38.990.000

Gì Cũng Rẻ ?
Giá tốt nhất so với các sản phẩm cùng loại trên Shopee!

Mã Giảm Giá Của Shop: 10% GIẢM

Protection: Bảo hiểm Thiết bị di động New Learn More

Vận Chuyển: Vận Chuyển Tới Huyện Ba Vì
Phí Vận Chuyển Không hỗ trợ

ROm: 128GB 256GB 512GB

Color: Red Green Black White

Số Lượng: 1 — 2232 sản phẩm có sẵn

Shopee Đảm Bảo — 3 Ngày Trả Hàng / Hoàn Tiền

phuvian
Online 23 Phút Trước

Đánh Giá 522 — Tỉ Lệ Phản Hồi 15% — Tham Gia 3 năm trước
Sản Phẩm 80 — Thời Gian Phản Hồi trong vài giờ — Người Theo Dõi 8k

Figure 1.1: Illustration of a product on e-commerce website

Image taken from Shopee at 01/05/2022

As just analyzed, from the perspective of a single customer, the comments of other buyers influence the buying decision process. Consequently, it is critical to investigate and comprehend clients' viewpoints through their thoughts and opinions. Organizations and businesses in the industry are more interested in learning what their customers or the general public think about their products and services, and so shifting their thinking from a product-centric to a customer-centric perspective. Making the decision to continue in customer sentiment and opinion analysis from online reviews not only helps businesses better understand the buying decision-making process of customers as well as their needs, but it also assists them in monitoring their brand, making improvements to their products or services, and developing more effective marketing strategies. Manual approaches, however, are prohibitively time-consuming and expensive, and as a result are not practicable due to the sheer volume of data. This has opened the door to automated approaches, and it is critical to develop an autonomous computational method for analyzing opinions hidden inside unstructured texts in order to bring the sentiment analysis field into the spotlight.

1.2 Problem Statement

1.2.1 Sentiment Analysis

Sentiment Analysis automates the extraction or classification of sentiment from opinions. According to Liu[7], an opinion can be described as a quintuple $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$, where

- e_i: An entity e_i is a product, service, topic, issue, person, organization, or event.
- a_{ij}: An aspect a_{ij} is a part or an attribute of entity e_i.
- h_k: The opinion holder.
- t_l: The time when the opinion is expressed by the opinion holder.
- s_{ijkl}: The sentiment expressed as positive, negative or neutral about aspect a_{ij} of entity e_i at time t_l given by opinion holder h_k

The objective of sentiment analysis is now can be defined as: Given an opinion document d, discover all opinion quintuples $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ in d.

In general, three degrees of sentiment analysis have been carried out. The task at the document-level sentiment classification level is to determine if an entire opinion document exhibits a positive or negative attitude by categorizing it as such in one of two ways. It is not, however, relevant to documents that evaluate or compare many entities at the same time. The aim at the sentence-level is to examine the sentences and assess whether or not each statement represents a good, negative, or neutral attitude about the subject matter in question. However, opinion mining at the phrase level is sometimes insufficient, because each opinion can refer to more than one aspect, and a specific aspect is inferred rather than a generic or entity aspect, as is the case in many instances. When it comes to applications, classifying opinion texts at the document or sentence level is frequently inadequate because it does not identify opinion targets or assign attitudes to such objectives. Even if we assume that each document examines a single thing, a document expressing a good view on the entity does not necessarily imply that the author holds positive attitudes about all aspects of the entity in question. A negative opinion document, on the other hand, does not necessarily imply that the author is negative about everything. It is necessary to identify the features and establish if the attitude is good or negative for each one in order to conduct a more thorough examination. As a result, we require a more in-depth examination of aspect-level sentiment, which is referred to as aspect-based sentiment analysis.

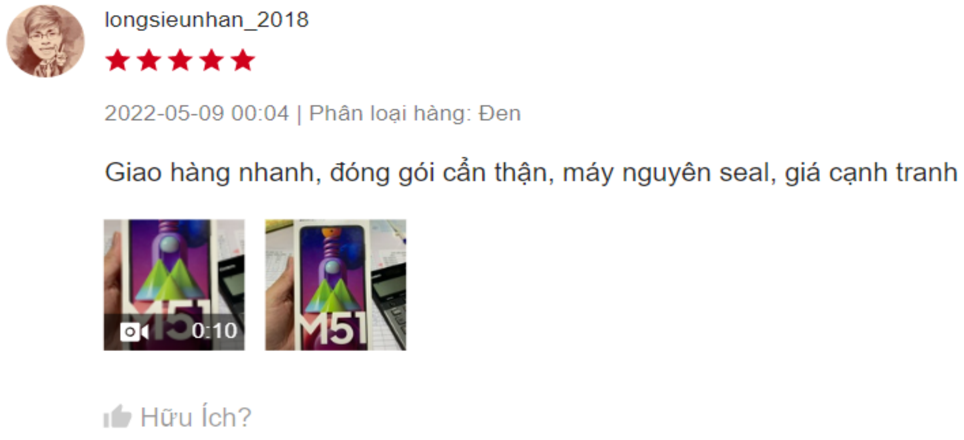
1.2.2 Aspect-Based Sentiment Analysis

Aspect-Based Sentiment Analysis (ABSA), the third level of sentiment analysis, is a technique for analyzing attitudes toward specific aspects of a situation, which helps opening a thorough understanding of and determine attitudes in relation to many facets of the subject.

In the words of Zhang et al. [11], for ABSA problem, the two main components are target and sentiment. The target can be an entity e_i, or an aspect of the entity itself, which can be described as either an aspect category c or an aspect term a. While, the sentiment include the opinion term o and the sentiment polarity p. ABSA's primary research focus is on these four sentiment elements:

- Aspect category c denotes a unique attribute of an entity and is expected to fall within a predefined category set C for each specific domain of interest.
- Aspect term a denotes the opinion target that appears explicitly in the given text, for example, "diapers" in the sentence "Soft, good quality diapers". When an implicit target is specified, a special aspect term known as "null" is used. When expressing an opinion about an entity, the special aspect "general" is now used.

- Opinion term o is the word or phrase given by the opinion holder to convey their feelings about the target. For the example "Soft, good quality diapers" the opinion term is "Soft, good quality".
- Sentiment polarity p denotes the sentiment's orientation toward an aspect category or term, which is positive, negative, or neutral.



longsieunhan_2018

★★★★★

2022-05-09 00:04 | Phân loại hàng: Đen

Giao hàng nhanh, đóng gói cẩn thận, máy nguyên seal, giá cạnh tranh

"Fast delivery, carefully packed, sealed machine, competitive price"

Giao hàng nhanh, **đóng gói** cẩn thận, ***máy*** nguyên seal, **giá** cạnh tranh

Figure 1.2: Example of comments about product in e-commerce website

Image taken from Shopee at 01/05/2022.
Bold italic word indicates entity, bold words indicate aspect, while regular words indicate opinion terms which imply positive polarity.
Users can also see opinion time and holder (in form of holder's username) and polarity intensity level (in form of stars)

Thus, the ABSA problem can be separated into two sub-tasks: aspect extraction and sentiment categorization of aspects of the problem. The first sub-task attempts to identify all characteristics that have been examined, while the second sub-task attempts to determine which sentiment polarity each aspect belongs to.

1.2.3 Thesis Scope

This thesis tackle single ABSA task classified as Aspect Sentiment Classification (ASC). An aspect sentiment classification model tries to figure out how someone feels about a certain aspect of a entity in a sentence. According to Zhang et al. [11], ASC problem position in general and this thesis problem position in detail in overall ABSA situation and relationship with other ABSA's problems can be illustrated in Figure 1.3.

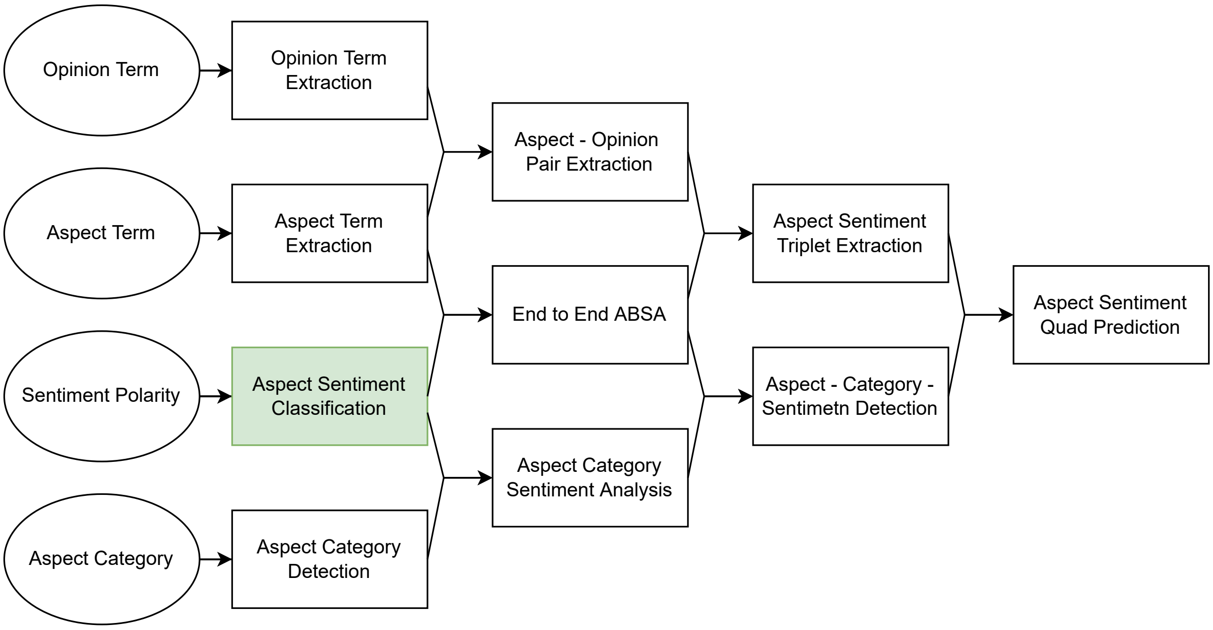

Opinion Term
Opinion Term Extraction
Aspect Term
Aspect Term Extraction
Sentiment Polarity
Aspect Sentiment Classification
Aspect Category
Aspect Category Detection
Aspect - Opinion Pair Extraction
End to End ABSA
Aspect Category Sentiment Analysis
Aspect Sentiment Triplet Extraction
Aspect - Category - Sentimetn Detection
Aspect Sentiment Quad Prediction

Figure 1.3: The relation between four sentiment elements, their corresponding tasks and compound ABSA tasks

The thesis solves the problem within the following constraints:

- The aspect is instantiated as aspect category. The category set is *price, service, delivery, performance, authenticity, hardware, accessories, appearance* for Technology domain, while the Mother & Baby category set includes *price, service, delivery, authenticity, safety, quality*.
- The sentiment that must be extracted does not include their term, but rather the polarity. Positive and negative polarity are considered here (not include neutral). $+1$ denotes positive polarity, -1 denotes negative polarity, and 0 denotes the unmentioned aspect.
- Focusing on ABSA elements as proposed above, opinion holder h_k and opinion time expressed t_l are not taken into account.

System must be able to recognize the most often discussed features of the entity (e.g., “quality”, "delivery"), as well as the sentiment associated with each aspect, i.e., whether the opinions are more positive or negative. Aspect recognition is considered as the preceding problem of sentiment classification and then outside the scope of this study. Consider the following example: *"Bỉm mềm, chất lượng tốt, hút được nhiều nhưng* *giao hàng khá chậm"* ("Soft, good quality diapers, absorb a lot but delivery is quite slow"). "Quality, positive" and "delivery, negative" are examples of aspect-sentiment tuples that must be determined by the system.

1.3 Contributions and Structure of the Thesis

Chapter 1 has outlined the dynamics of the problem from the actual demands of the problem from the standpoint of customers as well as businesses on e-commerce platforms.

With the challenge posed above, the thesis investigates the strategies to solve the ABSA problem, that is the sentiment analysis problem at the deepest level. The thesis has given the definition of SA and ABSA problems as well as the limitation of the scope that the thesis addreses.

From that foundation, this thesis contribution can be briefly described as follows:

- Provide a comprehensive multi-label classification system to perform the second sub-task of ABSA.
- Implement LDA generation model, but with improvements to be able to define the convergent aspects according to some pre-added seed words, thereby flexibly applying the subjective knowledge domain in the assessment and position of aspects and make active contributions to the model in order to better orient the model.
- Improving the results for the ABSA problem from the previous study on ABSA for Vietnamese e-commerce data, which was published at The 13th IEEE International Conference on Knowledge and Systems Engineering (KSE 2021) with the title "Aspect-Based Sentiment Analysis Using Mini-Window Locating Attention for Vietnamese E-commerce Reviews".

The remaining of the thesis respectively presents problems as follows:

- Chapter 2 describe the orientation and methodology of several studies that have been conducted in relation to this subject.
- Chapter 3 present the thesis directions for the ABSA problem that are relevant to the dedicated areas (Technology and Mother & Baby in detail).
- Chapter 4 details the experiments that were performed in order to calculate the statistics of all methods considered in each of the model's components.

- Conclusions section brings the issue to a close and identifies some directions to upgrade and improve the work in the future.

Chapter 2

Related Works

2.1 Current approaches to ABSA problem

As a result of its numerous uses in providing vital details on various parts of a sentence or document, ABSA has been extensively investigated in a number of different languages. Hu and Liu conducted the initial research and launched ABSA into the market, with the goal of determining product features and aspects that had been commented on by the reviewers. Brun et al. used the term frequency to manually develop features in a computer-assisted design environment.

In recent years, neural network-based systems have emerged as the most popular solution for the ABSA problem, owing to the fact that these systems can be taught from beginning to end and automatically learn crucial properties. Using target-dependent sentiment classification, Tang et al. create target-specific long- and short-term memory models for target-dependent sentiment classification. He et al. exploited attention mechanisms to obtain aspect-specific representations of sentences, based on the intuition that different sections of the sentence play different roles for different aspects of the sentence. Sentic LSTM was proposed by Ma et al. as an expansion of the long-short term memory (LSTM) network. Nguyen and Shirai developed a recursive neural network strategy to enrich the representation of the target aspect by including syntactic information into the network architecture.

The ABSA problem with Vietnamese textual data has been addressed by a variety of methods in recent years in Vietnamese, including the BRNN-CRF, Random Forest architecture, Support Vector Machine-based model, Semantic Relation Analysis, semi-supervised learning, etc.

This thesis examines the ABSA survey conducted by Zhang et al. [11]. In general, the aspect can be instantiated as either an aspect term or an aspect category, resulting in two correspoding ASC problems. The scope of this thesis is sentiment categorization based on aspect categories. In fact, some works simultaneously consider and address these two subtasks using the same model. Early ASC systems typically rely on manually designed attributes. In recent years, deep learning-based ASC has garnered significant interest, and a number of neural network-based models have been proposed and resulted in significant performance improvements, such as TC-LSTM, which employs relatively simple strategies such as concatenation to fuse aspect information with sentence context. Widespread usage of the attention mechanism to obtain aspect-specific representations. Attention-based LSTM with Aspect Embedding (ATAE-LSTM) is a model that appends the aspect embedding to each word vector of the input sentence for computing the attention weight, and a sentiment-specific aspect-specific sentence embedding can be computed appropriately. The subsequent techniques create more complicated attention mechanisms to learn better aspect-specific representations; for example, IAN learns interactively attention in the aspect and sentence, and generates representations for each individually. Other network architectures, such as the CNN-based network, memory network, and gated network, have been researched for supporting the attention mechanism.

Topic modeling has been widely utilized as a foundation for extracting and grouping aspects [2]. The LDA model specifies a Dirichlet probabilistic generative process for document-topic distribution; in each document, a latent aspect is selected based on a multinomial distribution and a Dirichlet prior. Then, given an aspect, a word is extracted based on a second multinomial distribution governed by a second Dirichlet prior. This thesis will introduce the LDA generation model in more detail in the following section. According to Poria et al. [9], existing works utilizing these models include the extraction of global aspects and local aspects, key phrases extraction, the rating of multi-aspects and the summarization of aspects and sentiments.Maximum-Entropy was used to train a switch variable based on the POS tags of words, which was then used to distinguish between aspect and sentiment words. DF-LDA is a semi-supervised model that permits the user to specify must-link (between terms of the same topic) and cannot-link (between two terms differentiated by topic) constraints.

2.2 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA), first introduced by Blei et al. [1] is a generative probabilistic model. In the LDA model class, you can define a series of imaginary topics, each of which is represented by a set of words. Basic concept is that documents are represented as random mixes over latent topics, with each subject being described by a distribution of words over the corpus.

2.2.1 Key Terms and Notations

- $word$ is the most basic unit of LDA. One word is identified by an index in the dictionary has value varies from $1, 2, ..., V$
- $document$ is a set $\mathbf{w} = (w_1, w_2, ..., w_N)$ which contains N words.
- $corpus$ is a set of M documents denoted by $\boldsymbol{D} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M\}$
- α is the parameter of the topic's Dirichlet a priori distribution for each text.
- β is the parameter of the topic's Dirichlet a priori distribution for each word.
- θ_i is the distribution of topic with respect to the i^{th} document.
- ϕ_k is the distribution of word with respect to the k^{th} topic.
- z_{ij} is topic of j^{th} word for i^{th} document.
- w_{ij} is the index in the vocabulary of the i^{th} word of the i^{th} document.

2.2.2 Generation Process

In order to derive subjects from the corpus, we have to design a method in which texts are generated in accordance with a technique that is both inferrable and reversible.

The probability distribution of the text is generated as a random mix of topics, where each topic is determined by the distribution over all words. LDA assumes a generation process for a corpus D consisting of the M documents is as follows:

1. For each document select the document length $N \sim \mathbf{Poisson}(\zeta)$
2. Select the matrix $\theta \sim \mathbf{Dir}(\beta)$ so that the θ_i is the topic distribution of the i^{th} document. The α parameter is usually a k-dimensional sparse vector with the majority of components being 0. Each dimension of α is specific to a topic.

3. Select ϕ to represent the distribution of words by topic. Similar to α and β, ϕ is also a sparse vector k-dimension which each of one dimension represent a topic. The distribution parameters are chosen to be sparse vectors so that each topic can only be explained by a small group of words belonging to that topic.

4. For each word w_{ij} in the i^{th} text and the j^{th} position in the text:

   (i) Select a topic distribution $z_{ij} \sim \textbf{Multinomial}(\theta_i)$.

   (ii) Select a word w_{ij} in the text i^{th} and position j^{th} .

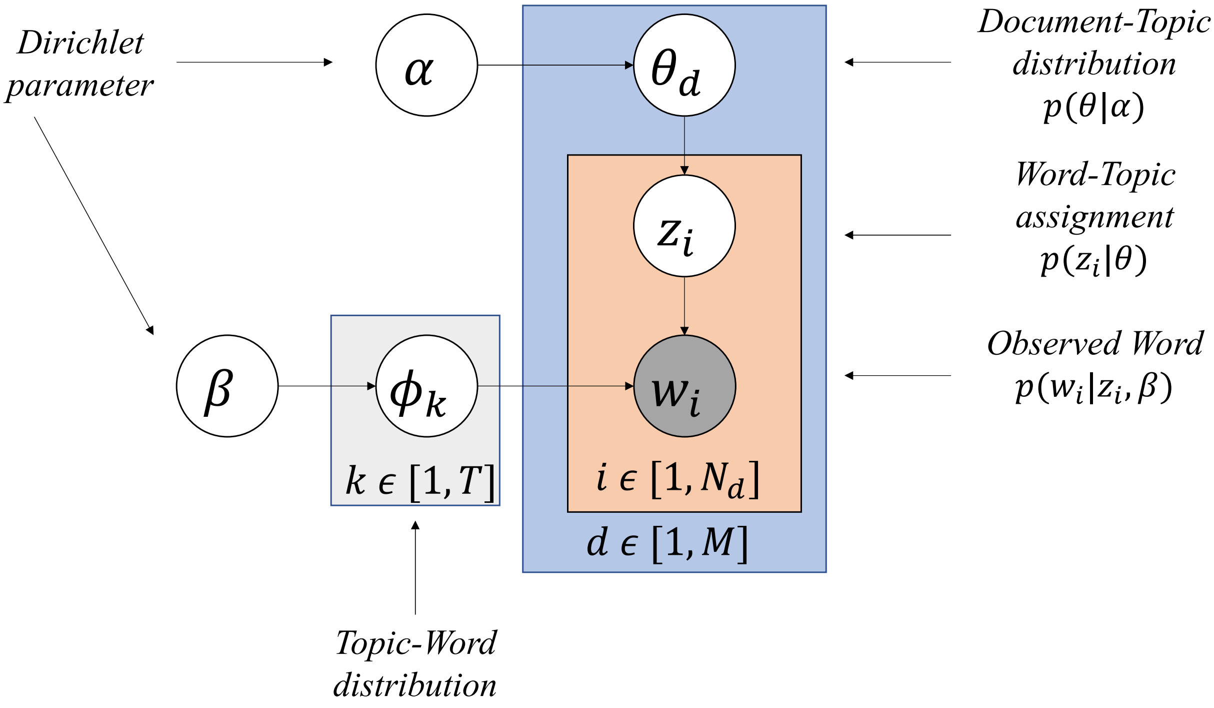


Figure 2.1: LDA graphical notation

We assume the number of hidden topics equal to k corresponding to the dimensionality of the known Dirichlet distribution, and the probability of the word being parameterized by a matrix $\beta \in \mathbb{R}^{K\times V}$ for each element β_{ij} characterizes the probability distribution of the word j^{th} for the topic i or $\beta_{ij} = p(w_j = 1|z_i = 1)$. Then we will fix these parameters.

Then we will calculate the probability density function of topics for each document after knowing the parameter α according to the Dirichlet distribution formula:

$$f(\theta;\alpha) = \frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma\left(\alpha_i\right)}\prod_{i=1}^{K}\theta_i^{\alpha_i-1} \tag{2.1}$$

Then the general probability distribution of mixed topic θ with set N topic z and N

from w with known parameters α, β is:

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{i=1}^{N} p\left(z_n \mid \theta\right) p\left(w_n \mid z_n, \beta\right) \tag{2.2}$$

The component $p(\theta|\alpha)$ is the mixed probability distribution of the topic corresponding to the document when the Dirichlet distribution parameter α is known in advance. Rest of the right hand side $\prod_{i=1}^{N} p\left(z_n \mid \theta\right) p\left(w_n \mid z_n, \beta\right)$ is the probability distribution when the mixed topic distribution and the Dirichlet distribution parameter β are known in advance. If we take the marginal probability of a document by integrating with respect to θ and sum all the z we get:

$$p(\mathbf{w} \mid \alpha, \beta) = \int p(\theta \mid \alpha) \prod_{i=1}^{N} \sum_{z_n} p\left(z_n \mid \theta\right) p\left(w_n \mid z_n, \beta\right) \mathrm{d}\theta \tag{2.3}$$

And finally we will calculate the probability of the entire text based on the marginal probability from each text.

$$p(\mathcal{D} \mid \alpha, \beta) = \prod_{d=1}^{M} \int p\left(\theta_d \mid \alpha\right) \prod_{i=1}^{N} \sum_{z_n} p\left(z_n \mid \theta\right) p\left(w_n \mid z_n, \beta\right) \mathrm{d}\theta_d \tag{2.4}$$

2.3 Semi-supervised SeededLDA

The original LDA will devote more attention to topics that are statistically significant and have a high frequency of terms. When topic models are fitted to a document collection, they implicitly use co-occurrence information at the document level to group semantically related words into a single topic. Due to the fact that the goal of these models is to maximize the probability of the observed data, they have a tendency to explain only the most obvious and superficial aspects of a corpus of observations. They effectively sacrifice performance on rare topics in order to do better modeling words that are frequently encountered. A semantically ambiguous topic is more likely to contain non-frequent terms that refer to different real-world topics that are mixed together in a semantically ambiguous topic. LDA models based on seeds have been proposed as a solution to this problem. The general concept behind them is to use seed information as prior knowledge to guide LDA and deliver topics that are more relevant to the user's interests than would otherwise be delivered. There are S seed sets in total, each of which contains seed words that are related to a specific topic.

This thesis applies SeededLDA based on approach by Jagarlamudi et al.[4] in incorporating lexical seeds into topics. SeededLDA improves both topic-word and document-topic distributions by using seed words. To improve the topic-word distributions, a model is created in which each subject likes to generate words that are linked to the seed set's words. To optimize the distribution of document-topic, the model is encouraged to choose document-level topics depending on the presence of input seed words inside the document. SeededLDA's generation process is illustrated by algorithm 1.

Algorithm 1: SeedingLDA Algorithm

```
for $k \leftarrow 1$ to $T$ do
    $\phi_k^r \leftarrow \mathbf{Dir}(\beta_r)$ ;                    // choose regular topic
    $\phi_k^s \leftarrow \mathbf{Dir}(\beta_s)$ ;                       // choose seed topic
    $\pi_k \leftarrow \mathbf{Beta}(1, 1)$;
end
for $s \leftarrow 1$ to $S$ do
    $\psi_k \leftarrow \mathbf{Dir}(\alpha)$;          // choose group-topic distribution
end
foreach document $d$ do
    $b \leftarrow \mathtt{Vector}(size : S)$ ;              // choose a binary vector
    $\zeta^d \leftarrow \mathbf{Dir}(\tau\vec{b})$ ;    // choose a document-group distribution
    $g \leftarrow \mathbf{Mult}(\zeta^d)$ ;                 // choose a group variable
    $\theta_d \leftarrow \mathbf{Dir}(\psi_g)$;
    for $i \leftarrow 1$ to $N_d$ do
        $z_i \leftarrow \mathbf{Mult}(\theta_d)$;                         // select topic
        $x_i \leftarrow \mathbf{Bern}(\pi_{z_i})$ ;                     // select indicator
        if $x_i = 0$ then
            $w_i \leftarrow \mathbf{Mult}(\phi_{z_i}^r)$;     // take from regular distribution
        else
            $w_i \leftarrow \mathbf{Mult}(\phi_{z_i}^s)$;        // take from seed distribution
        end
    end
end
```

Topic-Word Distributions Each topic k in the original LDA is defined by a multinomial distribution ϕ_k over words. While in SeededLDA two Multinomial distributions are used: one for normal words and another for seed words. A seed topic distribution uses a list of user-defined seed words to select words, whereas a regular topic distribution can use any word in the corpus (including seed words). Each seed topic is represented mathematically by a non-uniform probability distribution over the words in its set. Only the seed word sets are entered, and the model infers their probability distributions. Each document is a combination of a regular topic ϕ_r and its corresponding seed topic ϕ_s. The parameter π_k specifies the probability of encountering the word in either the seed words or the regular words distributions.

The processes are predicated on the assumption that exist a bijection from the seed topics to the regular topics. Real situations in which there are more subjects frequently result in the seed topics being duplicated. The first step in the above process is conducted to generate multinomial distributions for both seed and regular topics. The seed topics are chosen in such a way that their distribution produces only words from the pre-defined seeds. Following that, a topic for each words contained within the document is generated. After selecting topic, a (biased) coin (in form of a Bernoulli distribution) is flipped to determine whether to use the seed distribution or the standard topic distribution. Once the distribution is selected, a word is generated from it. It is critical to remember that, despite the existence of $2 \times T$ topic-word distributions in total, each document contains only T topics.

Document-Topic Distributions To improve the topic-word probability distribution, the model used seed words in the first step of the process. Following that, the model will make use of the seed words to increase the probability distribution of document topics. At this point, unlike step 1, the model no longer makes the same assumptions about the number of seed topics as it did about the number of regular topics. The seeds are now grouped together into groups, and these groups are then linked together by a Multinomial distribution over the regular topic, which is referred to as group-topic distribution.

The document-topic distribution generation is designed in a two-step approach to accommodate a flexible number of seed and regular topics while also tying the topic distributions of all the documents within a group together: In the first instance, the document-topic distribution is established in the following manner: the seed set (g for group) is sampled, and the resulting group-topic distribution (ψ_g) is utilized to produce the document-topic distribution (θ_d).

First, T topic-word distributions (θ_k) and S group-topic distributions (ϕ_s) are created. Then, for each document, a list of seed sets that are permitted for use with that document is generated. The binary vector $\vec{b}$ is used to represent the items on this list.

Due to the fact that this binary vector may be populated based on the document words, it is treated as an observable variable in the analysis. The binary vector $\vec{b}$, which shows which seeds are included in this document, defines the mean of a Dirichlet distribution, from which it is possible to sample a document-group distribution, ζ^d(step 3b). It is determined that the concentration of this Dirichlet is a hyperparameter, and so, $\zeta^d \sim \textbf{Dir}(\tau \sim \vec{b})$. A group variable g for this document is derived from the multinomial that results from this process. Because it groups documents that are likely to talk about the same seed set, this group variable helps to create a clustering structure among the documents in the collection. It is possible to select a document-topic distribution θ_d from a Dirichlet distribution with the group's topic distribution as the prior distribution after the group variable g has been drawn (step 3d). This stage guarantees that the subject distributions of documents within each group are consistent and related to one another.

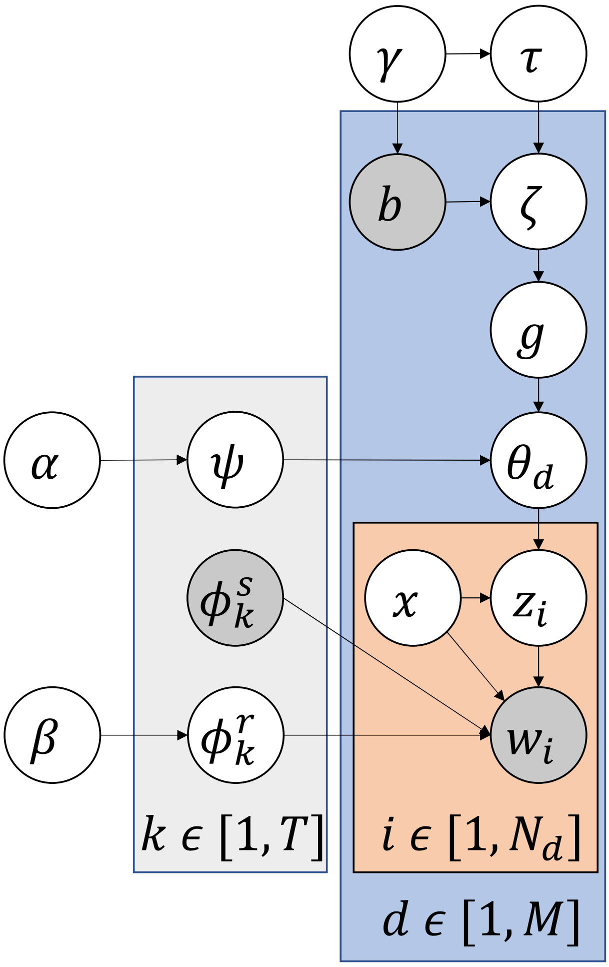


Figure 2.2: SeededLDA graphical notation

2.4 Classifiers used in Baseline Model

This section discusses the theory behind several of the statistical machine learning models implemented in the experiments, as well as some of their advantages and disadvantages.

2.4.1 K-nearest Neighbors

A simple supervised learning algorithm in machine learning is K-nearest neighbors, first introduced by Fix and Hodges [3].

In classification problem, a new data point's class is derive from k nearest data point from the training set. In general, this label can be calculated by the label of nearest data point $k = 1$ $(weights = uniform)$ or average weight of the nearest points $(weights = distance)$, or a relationship between these weight.

KNN is simple as only two criteria needed: value of k and the function to calculate distance. KNN does not require any training period because it does not derived any function from training data and only make real-time prediction when evaluating, so this make the algorithm run much faster, especially considering small dataset. New data can be also add anytime without any affection to the algorithm's accuracy.

The KNN algorithm has a unique property in that it is extremely sensitive to the distribution of data in a given area. In order to mitigate noise, this method is extremely sensitive. One example is when there is a single attribute in the data that is significantly greater than the rest, the distance between the points will be heavily influenced by this attribute, as shown in the graph below. Prior to running KNN, data normalization procedures can be applied in order to avoid this situation.

Its merits include being simple and easy to understand; having zero computing complexity during training; not requiring any assumptions about the distribution of classes; and performing well even when there are a large number of classes to categorize. Meanwhile, some disadvantages can be identified as being sensitive to noise (especially for small k - as described above), requiring a large amount of memory (when storing all of the data as well as the calculated distance), and becoming more time consuming as the number of data points or the dimensions of the data increase.

2.4.2 Logistics Regression

Logistics Regression (Logisitics Regression), in its basic form, uses a logistic function (e.g., sigmoid, tanh) to model a binary dependent variable. The prediction score of Logisitics Regression is calculated by formula:

$$f(x) = \theta(\mathbf{w}^T\mathbf{x})$$

where

θ: logistics function (e.g., sigmoid, tanh, etc.)

The cost function for Logisitics Regression is defined as:

$$L = \sum_{D} -ylog(y') - (1-y)log(1-y'))$$

where

D: dataset, containing of labeled tuples (x, y)

y: the label in a assigned example, which is either be 0 or 1.

y' is the predicted value, which is between 0 and 1, given features in x.

Logisitics Regression is one of the easiest machine learning algorithms as it is easy to implement, interpret, and very efficient to train. Traning a model with Logisitics Regression doesn't need high computation effort. Logisitics Regression also less prone to overfitting in a low dimensional dataset, and in context of a higher dimensional dataset, regularization can be used to avoid overfitting. Moreover, new data can be updated using Stochastic Gradient Descent.

But Logisitics Regression also has limitations as it only address linear separable data. For non-linear problems, transformation is required. Features used for training model should also be carefully extracted otherwise noise will make the probabilistic predictions may be incorrect. Logisitics Regression requires a large dataset and sufficient training examples for all the categories it needs to identify. Lastly, each training tuples must be isolated to all others, because relationship between any of them will make model give more importance to these relative examples.

2.4.3 Support Vector Machine

A Support Vector Machine model takes data points and outputs the hyperplane (a line in context of two dimensioal dataset) that best separates the classes. The best hyperplane is the one has largest distance to neares data point of each class. In other words, Support Vector Machine maximizes the margins from both class. With nonlinear data, additional dimensions will be required. Support Vector Machine can classify vectors in multidimensional space.

Support Vector Machine performs comparably well when the dissimilarity between classes is reasonable. It is more productive in spaces with greater dimensions. Support Vector Machine is also useful in situations where the number of dimensions exceeds the number of samples.

However, Support Vector Machine is memory systematic, which means it is unsuitable for use with large data sets. It performs poorly when the data set contains more noise and when the number of properties for each data point exceeds the number of training data samples.

2.4.4 Naive Bayes

Naive Bayes classifier based on applying Bayes' theorem. Bayes' theorem finds out the probability of an event if the occurence another event is probably knows, which is defined as:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|\neg A)\Pr(\neg A)} f(x)$$

where A and B are events.

Naive Bayes classifiers assumed all the features are independent and equally contributed to final result. Probability of feature y given a set of X features as $x_1, x_2, ..., x_n$ is calculated by formula:

$$P(y|x_1, x_2, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

We find the probability of all possible values of class y and choose the maximum, which can be expressed as:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \tag{2.5}$$

Although completely independent features are barely exist, in practical terms Naive Bayes is widely used since it's highly scalable, time-saving and suitable for categorical input variables.

Chapter 3

Proposed Models

3.1 Model overview

As described in Chapter 1, in order to be able to complete the ABSA task, the proposed model's primary goal is to answer the second sub-task of the ABSA problem. Specifically, the system should determine whether of two polarity labels - positive or negative - could be assigned to the associated aspect based on the information provided.

The proposed aspect-oriented sentiment analysis model is divided into four primary phases as shown in Figure 3.1: preprocessing, locating, representation and classification. The pipeline components can be briefly described as follows:

- Preprocessing: the first phase assists in cleaning and preparing data for categorization. Input is raw sentences while output is a set of sentences with each word tokenized.
- Locating: the second phase employs the method to identify words that have a high ability to determine which elements have a sentimental value.
- Representation: this thesis experimented with various ways of representing words with the goal of extracting useful features from the data, which include the main contribution in improving the overall model by using dictionary features (colored green in Figure 3.1).
- Classifiers: the final phase involves the application of numerous classifiers.

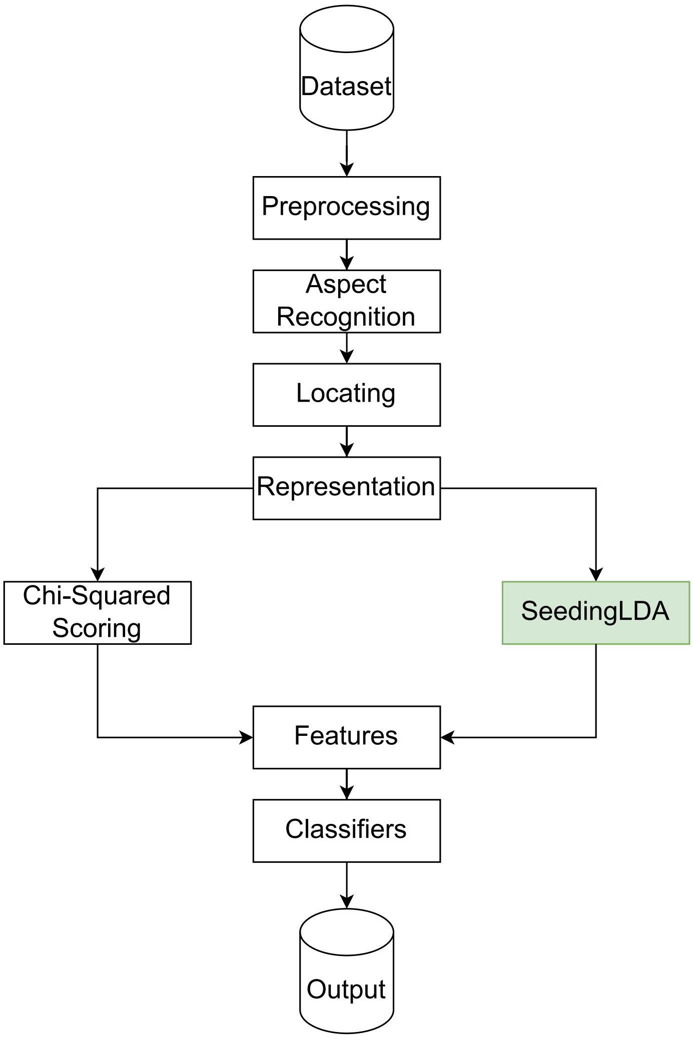


Figure 3.1: Model pipeline

3.2 Preprocessing

Preprocessing is the very first key stage in all natural language processing problems in general, and in ABSA problems in particular, because it changes the raw inputs into something that can be easily interpreted by a machine.

In the problem this thesis is working on, the input comes in the form of comments from e-commerce websites that are characterized as unstructured and lacking in standardization, as evidenced by the frequent occurrence of special letters, abbreviations, incorrect (missing/excess) spaces, misspellings, and other grammatical mistakes. If these noise points are not handled properly, they can have a negative impact on the model's performance. It is necessary to preprocess data in order to normalize it and remove noise. The following processes are taken to do this:

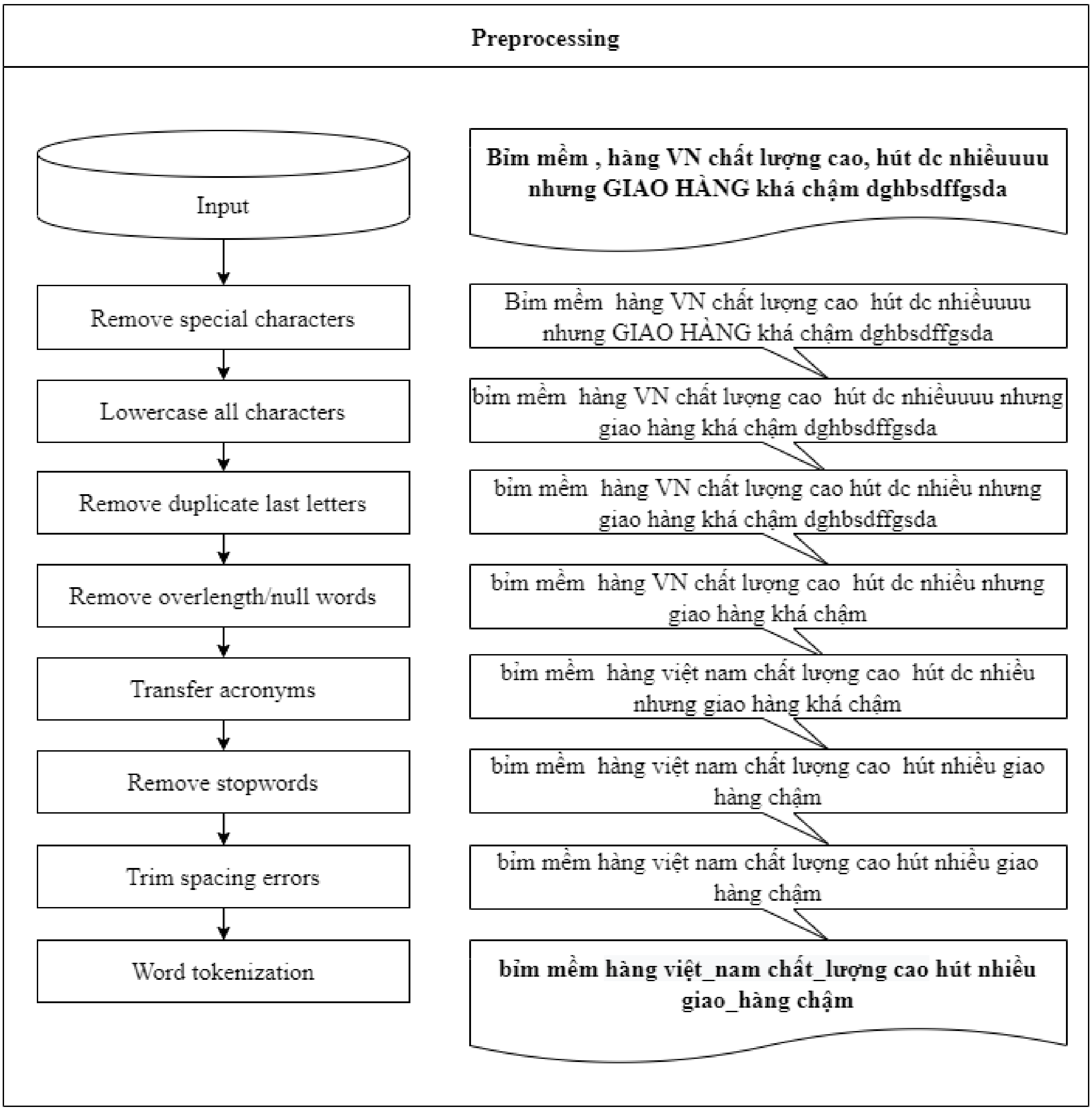

Preprocessing
Input
Bỉm mềm , hàng VN chất lượng cao, hút dc nhiềuuuu nhưng GIAO HÀNG khá chậm dghbsdffgsda
Remove special characters
Bỉm mềm hàng VN chất lượng cao hút dc nhiềuuuu nhưng GIAO HÀNG khá chậm dghbsdffgsda
Lowercase all characters
bỉm mềm hàng VN chất lượng cao hút dc nhiềuuuu nhưng giao hàng khá chậm dghbsdffgsda
Remove duplicate last letters
bỉm mềm hàng VN chất lượng cao hút dc nhiều nhưng giao hàng khá chậm dghbsdffgsda
Remove overlength/null words
bỉm mềm hàng VN chất lượng cao hút dc nhiều nhưng giao hàng khá chậm
Transfer acronyms
bỉm mềm hàng việt nam chất lượng cao hút dc nhiều nhưng giao hàng khá chậm
Remove stopwords
bỉm mềm hàng việt nam chất lượng cao hút nhiều giao hàng chậm
Trim spacing errors
bỉm mềm hàng việt nam chất lượng cao hút nhiều giao hàng chậm
Word tokenization
bỉm mềm hàng việt_nam chất_lượng cao hút nhiều giao_hàng chậm

Figure 3.2: Preprocessing steps

- **Step 1:** Special characters (including punctuation) have been omitted completely. Non-alphanumeric characters usually don't carry much meaning in general NLP problems. In the problem of the thesis, users can intentionally add some clusters of special characters in the form of emoticons, but these cases are not common and to make the model easier to handle characters all these characters are all removed.

- **Step 2:** Lowercase all characters. This is accomplished by converting the input text into the same case format as the output text, such that `"text"`, `"Text"`, and `"TEXT"` are all treated similarly. This may conceal the significance of the casing format, particularly in circumstances when users wish to stress their thoughts by capitalization. However, it will help similar words be considered the same and thereby help in increase the accuracy of Chi-Squared scoring calculation, which will be introduced in the latter section of the model.

- **Step 3:** Remove the last letters that are duplicated. In some circumstances, the double (or more) repeating letters at the conclusion are written on purpose to draw attention to the writer's goal; nevertheless, for the sake of this thesis, all of the duplicated letters are deleted for the sake of convenience during the next steps. Until further notice, the terms `"text"` and `"texttt"` are regarded synonymous.

- **Step 4:** Remove long words or null words. Every Vietnamese word has a maximum of 7 letters, so any word longer than 7 letters or shorter than 1 letter is discarded.

- **Step 5:** Transfer acronyms to the full form of them and translated from context. In natural language text, individuals frequently write the same entity differently. "Việt Nam", for example, may be spelled "vn", "vnam", or "việt nam". All of these spellings should be understood as refer to the same thing.

- **Step 6:** Delete stop words. A stop word is a commonly used word that the machine should avoid learning to save space and time processing. They do not act as meaningful criteria in our approach, so basically all of them were removed. However, some of the normal stop words may determine the sentiment, so we have to take an extra step manually to keep this kind of stop words.

- **Step 7:** Trim spacing errors which may be derived from the original data or generated as a result of the preceding preprocessing steps.

- **Step 8:** Word tokenization. Tokenization is the process of breaking down a phrase, sentence, paragraph, or even an entire text document into smaller units called tokens, which can be individual words or phrases. By tokenizing the text, certain analysis methods such as counting the number of words appearing, the frequency of the word, and so on can be used to more easily interpret the text's meaning.

3.3 Locating using Mini-Window Locating Attention

The proposed model implements Mini-Window Locating Attention from my previous work[6]. MWLA uses chi-squared rank as calculated above to weight the words in the comment, with a view to select out which word play the important role on determine aspect's sentiment.

Given a document D, chi-squared rank is estimated by equation:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where

χ^2 = Pearson's cumulative test statistic.

O_i = the number of observations of class i.

E_i = the expected count of observations of class i.

For each feature, a corresponding high χ^2 score indicates that the null hypothesis of independence (meaning the document class has no impact over the term's frequency) should be dismissed and the occurrence of the term and class are dependent, therefore the feature should be selected for classification. In other words, using this method remove the feature that are most likely autonomous of class and consequently unessential for classification.

This thesis use Scikit-learn[8] as it provide a *SelectKBest* class that can be used with various statistical tests, including Pearson's Chi-Squared test. It will rank the features with the statistical test and select the top k performing ones, which in this situation, indicates that these terms are deemed to be more relevant to the task than the others.

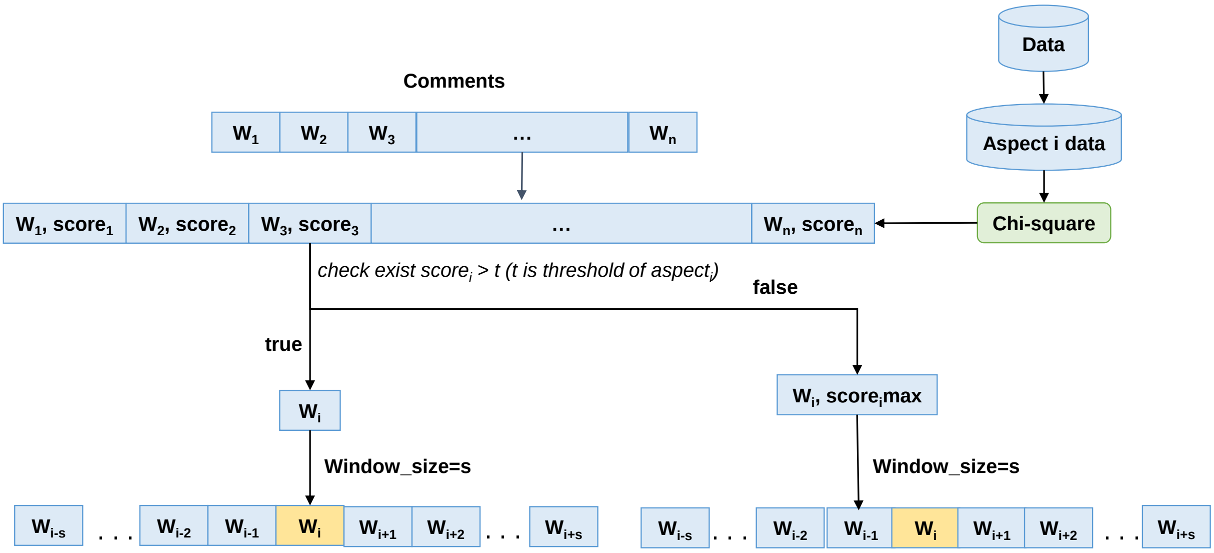

Data
Aspect i data
Chi-square
Comments
W1 W2 W3 ... Wn
W1, score1 W2, score2 W3, score3 ... Wn, scoren
check exist scorei > t (t is threshold of aspecti)
true
false
Wi
Wi, scoreimax
Window_size=s
Window_size=s
Wi-s . . . Wi-2 Wi-1 Wi Wi+1 Wi+2 . . . Wi+s
Wi-s . . . Wi-2 Wi-1 Wi Wi+1 Wi+2 . . . Wi+s

Figure 3.3: Mini-Window Locating Attention process

MWLA can be briefly described as follows:

- Assuming a sentence contains a set of word $W = \{w_0, w_1, w_2, ..., w_n\}$ and a set of aspects $A = \{a_0, a_1, a_2, ..., a_m\}$.
- Score $s_{i,j}$ is defined as χ^2 score of the word w_i in aspect c_j.
- For each aspect c_j, a threshold s_{c_j} is set, all the words which have lower score than that threshold will be discarded.
- If the threshold is larger than any score of corresponding word in sentence, the word with highest score will be selected.
- Take out s words (indicate size of the *window*) before and after the high-score word chosen above. This combination, or *window*, can be illustrated as:

  $\{w_{i-s}, w_{i-s+1}, ..., w_i, ..., w_{i+s-1}, w_{i+s}\}$

The results collected is helpful for classifier as MWLA removed the non-related words of each aspect and its sentiment; therefore, only relevant words extracted to put in learning model.

3.4 Representation

3.4.1 Chi-Squared

This thesis use word-level χ^2 as calculated above to weight the words in the vocabulary, and use that vocabulary to represent data. Then each aspect's vocabulary is filtered manually to reduce dimensions used for classfiers.

Some advantages of representing data using Chi-Squared points can be mentioned as follows:

- Taking advantage of the calculation process in the locating step.
- Easy to expand (when adding new words, just calculate the chi-squared score of that new word)
- Does not depend on user knowledge (user knowledge can be flexibly applied thanks to dictionary features - will be presented in the following sections)

3.4.2 Dictionary Features

For each aspect, a dictionary S_i is created using **guidedLDA**[1] library, according to the following procedure:

- Based on knowledge a set of self-selected words are produced as S_0. For example can use the words in *Example seed words* in the tables 3.1 and 3.2.

Table 3.1: First round seed words for Technology domain

Aspect	***Example seed words***
Price	xu, vô_địch, ưu_đãi, túi_tiền, tiết_kiệm, tiền, sale, rẻ, price, mua, mắc, khuyến_mại, khuyến_mãi, hời, giảm, giá_thành, giá_cả, giá, flashsale, đắt
Service	xử_lý, uy_tín, trả_lời, thờ_ơ, thô_lỗ, thiện_chí, thắc_mắc, thanh_toán, thái_độ, tương_tác, tư_vấn, tổng_đài, tin_tưởng, tận_tình, tận_tâm, shop, bố_đời, rep, phục_vụ, phong_cách, phản_hồi, nhân_viên, nhắn_tin, lừa, lịch_sự, khách_hàng, hỗ_trợ, hậu_mãi, giải_đáp, đánh_giá, dịch_vụ, chuyên_nghiệp, chu_đáo, chế_độ, chăm_sóc, cskh, cọc_cằn, bán_hàng, bảo_hành
Delivery	xộc_xệch, xấu, vận_chuyển, trễ, trầy_xước, thùng, tận_tay, tận, sớm, sơ_sài, shipper, ship, nhăn_nhúm, nhanh_chóng, nhanh, bọc, ẩu, xốp, nguyên_vẹn, móp, méo, lâu, kỹ, kĩ, hàng, giao, gửi, gọn_gàng, gói_hàng, gói, đúng_hẹn, đúng_giờ, đóng_gói, dự_kiến, cẩn_thận, cẩn
Performance	ứng_dụng, trơn_tru, rất mượt, phần_mềm, ổn_định, nhạy, nguồn, nóng, mượt_mà, mượt, mạnh, lỗi, lag, khởi_động, hoạt_động, hiệu_năng, giật, đứng, đơ, độ mượt, đáp_ứng, chức_năng, chậm
Authenticity	bảo_hành, bill, code, công_ty, chính_hãng, fullbox, giả, imei, LL/A, mã, mã_vạch, mác, niêm_phong, nghi_ngờ, nguyên_đai, nguyên_kiện, nguyên_vẹn, nhãn, QR, quét, seal, tem, vạch, VNA}
Hardware	thẻ_nhớ, sóng, sạc, rè, ram, pin, nghe, micro, mic, màn_hình, màn, loa_thoại, loa_ngoài, loa, chụp, camera, cảm_ứng, bộ_nhớ, âm_thanh

[1] https://github.com/vi3k6i5/guidedlda/

Table 3.1: First round seed words for Technology domain

Aspect	*Example seed words*
Accessories	tặng, tai_nghe, sạc, phụ_kiện, ốp_lưng, ốp, đầy đủ, củ sạc, case, cáp sạc, bao_da
Appearance	zin, xước, xinh_xắn, viền, vẻ_ngoài, trọng_lượng, trầy, trầy, thiết_kế, ọp_ẹp, ngoại_hình, nắm, mới, móp, mong_manh, mẫu_mã, mặt_lưng, màu, lõm, khung, hoàn_thiện, hình_thức, hầm_hố, đẹp, đầm, dễ_vỡ, cấn, cầm, build, bé, bắt_mắt, bao_đẹp

Table 3.2: First round seed words for Mother&Baby domain

Aspect	*Example seed words*
Price	xu, vô_địch, ưu_đãi, túi_tiền, tiết_kiệm, tiền, sale, rẻ, price, mua, mắc, khuyến_mại, khuyến_mãi, hời, giảm, giá_thành, giá_cả, giá, flashsale, đắt, đáng, deal, bình_dân
Service	xử_lý, uy_tín, trả_lời, thờ_ơ, thô_lỗ, thiện_chí, thắc_mắc, thanh_toán, thái_độ, tương_tác, tư_vấn, tổng_đài, tin_tưởng, tận_tình, tận_tâm, shop, bố_đời, rep, phục_vụ, phong_cách, phản_hồi, nhân_viên, nhắn_tin, lừa, lịch_sự, khách_hàng, hỗ_trợ, hậu_mãi, giải_đáp, đánh_giá, dịch_vụ, chuyên_nghiệp, chu_đáo, chế_độ, chăm_sóc, cskh, cọc_cằn, bán_hàng, bảo_hành
Delivery	xộc_xệch, xấu, vận_chuyển, trễ, trầy_xước, thùng, tận_tay, tận, sớm, sơ_sài, shipper, ship, nhăn_nhúm, nhanh_chóng, nhanh, nguyên_vẹn, móp, méo, lâu, kỹ, kĩ, hàng, giao, gửi, gọn_gàng, gói_hàng, gói, đúng_hẹn, đúng_giờ, đóng_gói, dự_kiến, cẩn_thận, cẩn, bọc, ẩu, xốp
Safety	bẩn, hăm, mẩn, date, dị_ứng, ghê, sử_dụng, hạn, hsd, đảm_bảo, hắc, khó_chịu, sờn, tràn, rách
Quality	mùi, tràn, thơm, chắc_chắn, chật_chội, chất_liệu, chất_lượng, vết, vận_động, đệm, may, êm, dáng, hằn, hôi, nhỏ, hút, nhạy, xệ, dày, đẹp, đều, phủ, lằn, mềm, mềm_mại, mịn_màng, mỏng, mùi, nhẹ, nhỏ, thô, khít, ôm, cũ, hút, thấm, mát, thoáng, cứng
Authenticity	bill, mã, code, chính_hãng, giả, công_ty, QR, vạch, niêm_phong, nhãn, mác, mã_vạch, tem, bảo_hành, quét

- Launch SeededLDA in i times (rounds) with parameters:
  - Initialize word generation times *seed_round* using SeededLDA.
  - Set iteration times *n_iter*.
  - Set the number of words taken after each round *n_top_word*.
  - Define $seed\_confidence$. Seed confidence indicates how much bias should be given to seed topic.
- After running each round, the *topic_word* in *n_top_words* are combined with the words of the set S_{i-1} to form the set S_i.

The dictionary set after the final round is hand-selected to remove duplicate words, words are too different from the aspect, etc. to evaluate the noise factor when generating the dictionary after many rounds. Illustration of the process of creating a dictionary after seeding is shown in figure 3.4

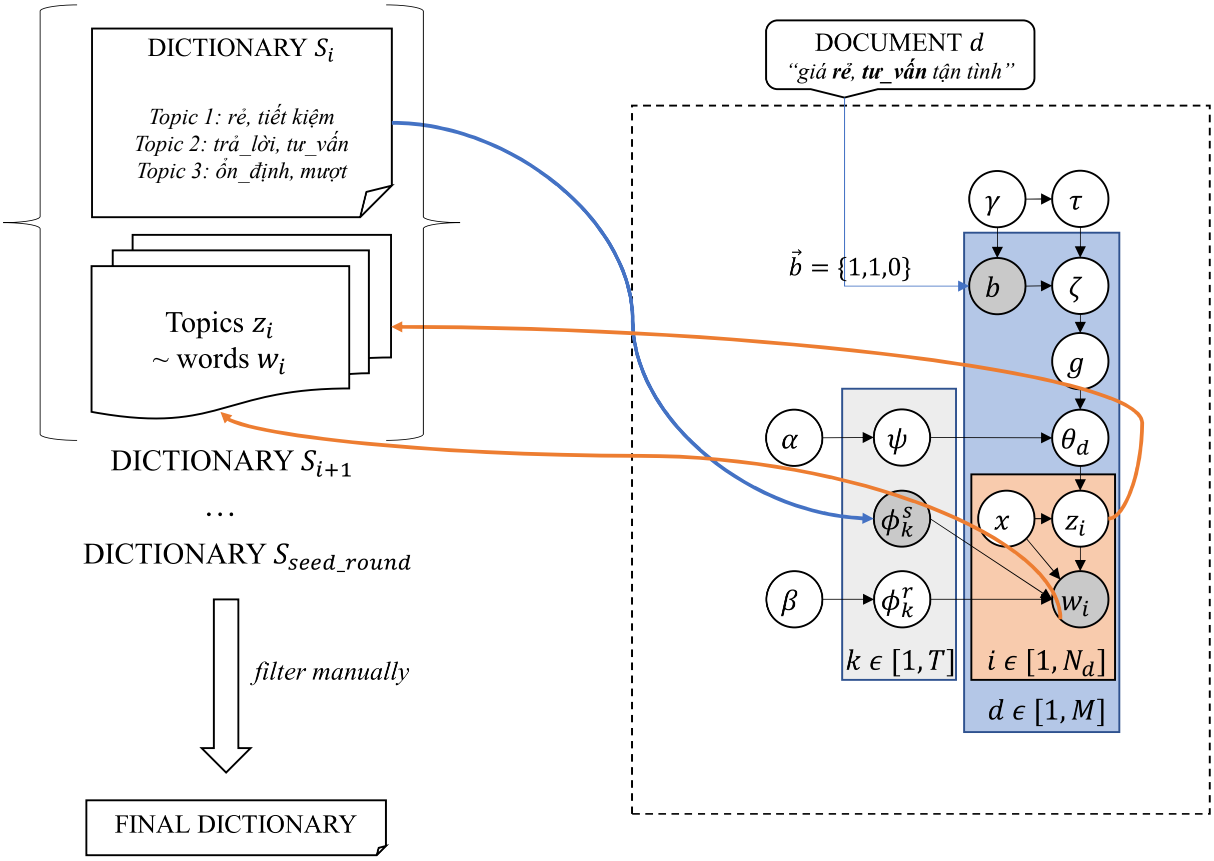


Figure 3.4: Dictionary Creating Process

During the tests the following parameters were fixed at the corresponding values:

- $seed\_round = [1...5]$

- $n\_iter = 1000$
- $n\_top\_word =$ *seed_round*$*10$ (round 1 takes 10 words, round 2 takes 20 words,...)
- $seed\_confidence = 1.1 -$ *seed_round*$/10$ (corresponding to 10% confidence reduction after each round)
- The Dirichlet parameters (α, β, ζ) are set to the default value of the **guidedLDA** library.

3.5 Classifiers

After the data has been processed by the MWLA, the output is a sentence devoid of irrelevant words. The words generated by SeedingLDA in each seed are used to create a new representation vector for the sentence expression with feature. The words generated by SeedingLDA are represented by onehot as a bag of words. After each sentence has been assigned a vector, these vectors are fed into a classifier to conduct the learning process. This thesis uses the basic statistical classifiers, with the theoretical background mentioned in the section 2.4. All of them are used to compare and select the baseline with detailed statistics in the subsection 4.2.1. Experimental results show that using LR is better than other models in the thesis situation.

Chapter 4

Experiments and Results

For the purpose of demonstrating the usefulness of SeedingLDA in recognizing specific elements of documents, the following experiments were carried out. In this section, this thesis will introduce experimental dataset, test scenarios, metrics and results that were employed, as well as some analytical findings. The source code of the deployed model can be viewed at Github[1]

4.1 Experiment Setup

4.1.1 Experimental Environment

Experimental environment of the desktop used in evaluation scenarios including:

- OS: Microsoft Windows 11 Pro
- Processor: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 2496 Mhz, 4 Core(s), 8 Logical Processor(s)
- RAM: 32GB
- System type: x64-based

Libraries (written in Python language) used include:

- `guidedLDA`: implements latent Dirichlet allocation (LDA) using collapsed Gibbs sampling.

[1] `https://github.com/LukeShrek/-SeededLDA-for-Sentiment-Classification`

- `guidedLDA` dependencies: `NumPy, pbr`
- `scikit-learn`[8]: `scikit-learn` is a Python module for machine learning built on top of `SciPy`. Scikit-Learn enables convenient access to a large number of different classification techniques, including the classifiers used in this thesis evaluation process.
- `scikit-learn` dependencies: `SciPy, joblib, threadpoolctl`
- `VnCoreNLP`[10]: `VnCoreNLP` is an NLP annotation pipeline for Vietnamese, providing rich linguistic annotations through key NLP components of word segmentation, POS tagging, named entity recognition (NER) and dependency parsing. This thesis uses `VnCoreNLP` for preprocessing tasks.
- `Pandas`: used for supporting data manipulations.

4.1.2 Experimental Data

In preparation to evaluate the effectiveness of the proposed model, this thesis conducted experiments on the dataset collected from Shopee[2] and Tiki[3]. This challenge necessitates the processing of text data, which comes in the form of comments on e-commerce sites. The dataset that was utilized was the Vietnamese E-commerce dataset, which was taken from my previous work[6]. Vietnamese E-commerce dataset was created by gathering 12240 comments on two well-known e-commerce platforms in Vietnam, Shopee and Tiki and it is divided into two data domains: retail and wholesale. Technology (as represented by a common smartphone device) and Mother & Baby are two notable materials (with a typical baby diaper product).

In order to label this data set, the data is traditionally included 6 aspects of Mother & Baby products (*pricing, service, delivery, safety, quality, authenticity, and authenticity*) and 8 aspects of Baby (*price, service, delivery, performance, hardware, authenticity, accessories, and appearance*). If there is a term or phrase that has the connotation of complaint in it, it is deemed negative for that element; if there isn't, it is considered positive. Table 4.1 and 4.2 covers the respective elements in depth, and also proposes some words/phrases that can be used to help distinguish between the aspects.

[2] `https://shopee.vn/`

[3] `https://tiki.vn/`

Table 4.1: Technology apects brief description

Aspect	***Description***
Price	Decrease/reduce/slash/low price are regarded as positive terms, increase/put up/raise/high price are regarded as negative terms.
Service	Prompt, efficient and enthusiastic help is considered positive, while no response, irresponsibility or negligent packing are deemed negative factors.
Delivery	The shipping process's overall quality, speed, and cost. If it is delivered quickly, carefully, and at a low cost, it is deemed positive; otherwise, it is considered negative.
Performance	It is deemed positive when product's processing speed is fast; nevertheless, it is deemed negative when a program has lag or latency in the middle of an application or when an application unexpectedly closes.
Authenticity	False or imitation products are deemed negative, while legitimately created products with identification signals such as an undamaged seal, a confirmed manufacturer's imei code, and so on are considered positive.
Hardware	The overall quality of the device's hardware, which includes the display, chip, battery, cameras, storage, RAM, and so on.
Accessories	Accessories that are fully given and of high quality are considered positive; low quality or missing accessories are considered bad.
Appearance	Product with a pleasing design, pleasing color, luxurious feel, and so on is considered good; product with scratches or an obnoxious design is deemed bad.

Table 4.2: Mother & Baby aspects brief description

Aspect	***Description***
Price	Decrease/reduce/slash/low price are regarded as positive terms, increase/put up/raise/high price are regarded as negative terms.
Service	Prompt, efficient and enthusiastic help is considered positive, while no response, irresponsibility or negligent packing are deemed negative factors.

Delivery	The shipping process's overall quality, speed, and cost. If it is delivered quickly, carefully, and at a low cost, it is deemed positive; otherwise, it is considered negative.
Safety	Product with an obvious expiration date that has not expired and is still safe to use is considered positive, while product that has expired and causes allergies or rashes is deemed negative.
Quality	The softness, absorbency of the diapers, the scent, and other aspects of the product that the customer has experienced are all considered. Positive experiences can be aromatic, soft, pleasant, and absorbent, whereas unpleasant experiences can be unfamiliar scents, harsh, and spilled, etc.
Authenticity	False or imitation products are deemed negative, while legitimately created products with identification signals such as an undamaged seal are considered positive.

4.1.3 Baseline Methods and Experiment Scenarios

It has been necessary to compare various combinations of classifiers and different representation methods in order to determine which one is the most effective in this thesis particular situation. The process of evaluation is:

(1) Logistics Regression, Naive Bayes, K-nearest Neighbors and Support Vector Machine combine with MWLA to find out which statistical classification method performs best.

(2) Model with best performance in (1) is considered as baseline. LDA seeded features are now considered by using the dictionaries that outputs by 1-5 seeding times in turn with varying confidence without any manual reviews, and the dictionary with further filtering. All of these dictionaries are used to observe the growth and efficiency of SeedingLDA after seeding times.

(3) General comparison of models with dictionary features performed best in (2).

4.1.4 Evaluation Metrics

Precision Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

where

TP = True Positive - correctly predicted positive values which means that the value of actual class is 1 and value of predicted class is also 1.

FP = False Positive - actual class is 0 and predicted class is 1.

Recall Recall is the ratio of correctly predicted positive observations to the all observations in actual class. High recall is synonymous with the low false negative rate.

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

where

FP = False Negative - actual class is 1 and predicted class is 0.

F1-score The F1 is a way of combining the precision and recall of the model. The higher the F1 score the better, with 0 being the worst possible, which means the precision or recall is zero; and 1 being the best, indicating perfect precision and recall. F1-score is defined as the harmonic mean of the model's precision and recall.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.3}$$

Macro-average Performance Macro Average Performance are used to evaluate multi-label classification model. A macro-average will compute the metric independently for each class and then take the average. Macro-average for Precision, Recall and F1-score is calculated by the formulas as shown in the Equations from 4.4.

$$P_{macro} = \frac{\sum_i P_i}{N} \qquad R_{macro} = \frac{\sum_i R_i}{N} \qquad F1_{macro} = \frac{\sum_i F1_i}{N} \tag{4.4}$$

4.2 Experiment Result and Analysis

4.2.1 Baseline Choosing

Table 4.3 illustrates performance of multiple statistical classifiers when combined with MWLA. The numbers in bold illustrate the highest figures obtained. Meanwhile, the numbers in italics illustrate the data of the selected model as the baseline.

In terms of macro-F1, the LR classifier outperforms the other models at least 0.055 when used with MWLA for negative label, and just 0.007 lower than the best performed model. In general, however, this figures demonstrate that the model without dictionary features has not fared very well in this situation (worst performed classifier only achieve approximately 0.279 for macro-F1). The reason for this can be attributed to the fact that the data is not balanced and that the negative labels are insufficient, resulting in the model not being recognized.

Based on these analysis, the thesis used Logistics Regression for the remaining evaluations.

Table 4.3: MWLA with statistical classifiers

Label	Metric	***Support Vector Machine***	***K-nearest Neighbors***	***Logistics Regression***	***Naive Bayes***
Positive	**macro-p**	0.962	0.956	***0.972***	0.958
	macro-r	**0.989**	0.925	*0.965*	0.984
	macro-f1	**0.975**	0.937	*0.968*	0.970
Negative	**macro-p**	0.583	0.393	***0.639***	0.559
	macro-r	0.440	0.387	***0.558***	0.426
	macro-f1	0.493	0.279	***0.548***	0.456

4.2.2 Seeded Dictionary Performances

Table 4.4 describes the Macro-average metrics when integrating the Dictionary Features into the baseline model. The figures highlighted in bold represent the highest results obtained in these experiments.

In general, the data has improved compared to the baseline model when the presence of dictionary features is present. Regarding the data of positive labels, in general, it has been relatively high right from the first seed, so the change and improvement are not observed too clearly. However, when looking at the negative label we can see an improvement in all three metrics, with 1.2%, 6.3% and 7.4% respectively. This growth, compared to the baseline model, is 12.9%, 11.1% and 13.5%, respectively.

		Baseline	*0*	*1*	*2*	*3*	*4*	*5*	*Filtered*
Positive	**macro-p**	0.972	0.973	0.973	0.974	0.974	0.976	0.976	**0.977**
	macro-r	0.965	0.961	0.962	0.962	0.963	0.965	0.969	**0.969**
	macro-f1	0.968	0.966	0.967	0.967	0.968	0.970	0.972	**0.972**
Negative	**macro-p**	0.639	0.750	0.759	0.758	0.768	0.755	0.762	**0.768**
	macro-r	0.558	0.593	0.600	0.605	0.611	0.662	0.656	**0.669**
	macro-f1	0.548	0.597	0.606	0.608	0.617	0.669	0.671	**0.683**

Table 4.4: Macro-F1 for Dictionary Features generated by SeededLDA

For a closer look, figure 4.1 and 4.2 plots these data on a line chart. The general trend depicted by the trendline indicates that the model outcomes increase steadily with each dictionary generation, despite the possibility of seeds that produce poorer results after round seeding. In particular, when utilizing a vocabulary that has been filtered to remove duplicates and fewer relevant terms, the results of the model are consistently at the highest level.

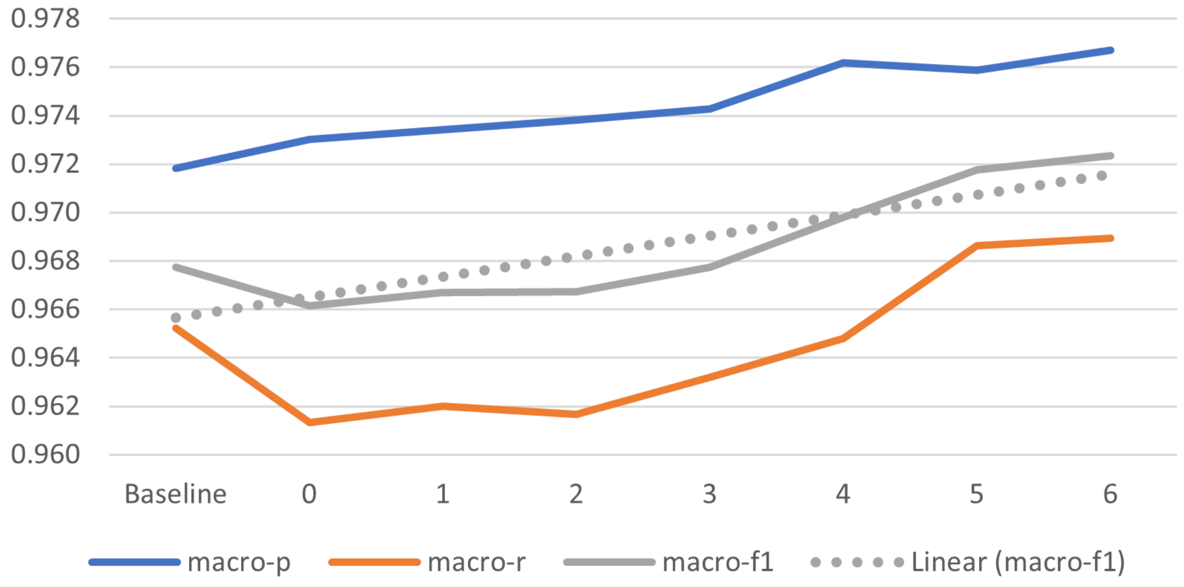


Figure 4.1: Trends observed when using SeededLDA in terms of positive label

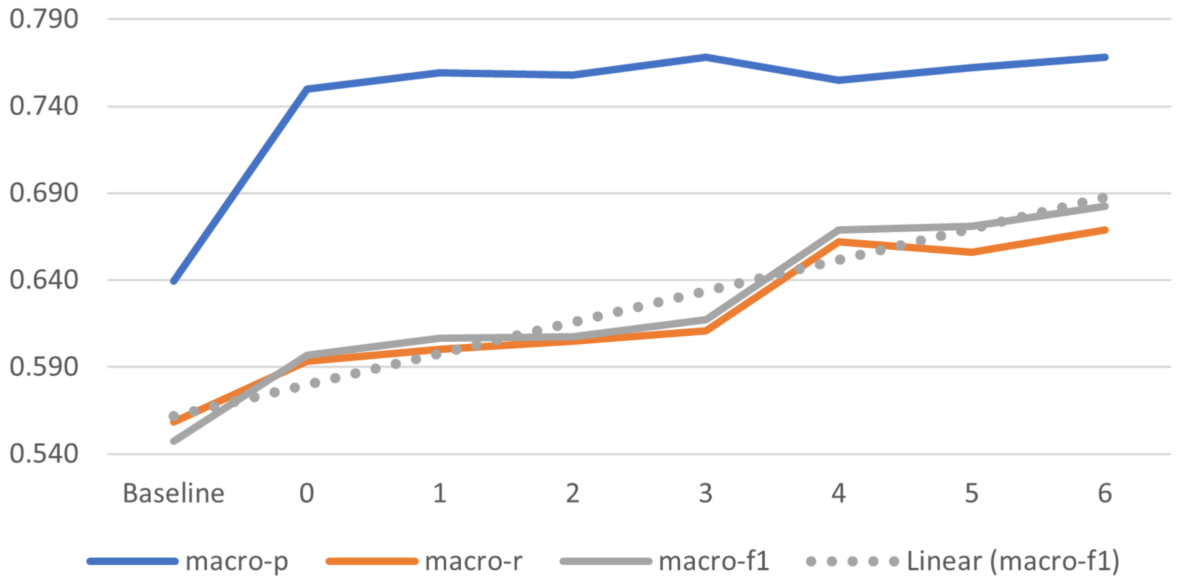


Figure 4.2: Trends observed when using SeededLDA in terms of negative label

4.2.3 Overall Performance using Filtered Dictionary Features generated by SeededLDA with baseline model

When the best model (Filtered Dictionary using SeededLDA + MWLA + LR) is evaluated, the data shown in the table 4.5 are achieved. The number in bold indicates the label that corresponds to the highest possible score. The best model achieves the greatest F1 score possible, reaching up to 99.3% for positive label and 90% for negative label.

These results reveal the validity of the model proposed in this thesis. In order to have a more in-depth analysis of the results obtained, however, the following section of the thesis will provide brief comments on the most common errors encountered; from there, future directions will be suggested.

4.2.4 Error Analysis

Confusing Generated Words Although based on probability calculation, it is inevitable that the model generates words that are not related to the aspect or cannot identify the aspect due to different meanings in different contexts. An example is shown in the table 4.6.

It can be seen that after generated, besides the words related to the aspect, there are also confusing words (the context cannot be determined about which aspect or the word has no meaning to define the aspect) printed in italics, or words that refer to a completely

	Positive			Negative		
	P	**R**	**F1**	**P**	**R**	**F1**
price	0.986	0.978	0.982	**0.978**	0.667	0.793
service	**0.989**	0.996	**0.993**	0.947	**0.857**	**0.900**
delivery	0.963	0.966	0.965	0.524	0.500	0.512
performance	0.978	0.967	0.972	0.500	0.600	0.545
authenticity	0.973	0.929	0.950	0.434	0.286	0.345
hardware	0.980	0.963	0.972	0.656	0.778	0.712
accessories	0.969	0.861	0.912	0.545	0.857	0.667
apperance	0.977	**0.997**	0.987	0.971	0.805	0.880

Table 4.5: Best performed model P, R, F1 score for Tech domain in all aspects

ứng_dụng, trơn_tru, mượt, phần_mềm, ổn_định, nhạy, nguồn, nóng, mượt_mà, mượt, mạnh, lỗi, lag, khởi_động, hoạt_động, hiệu_năng, giật, đứng, đơ, độ, mượt, đáp_ứng, chức_năng, *chậm*, **giao**, **màu**, *hôm*, *gọi*, *xong, tr, mấy, đều*, **tiền**, *xanh*

Table 4.6: Sample dictionary (performance - Technology) after seeding

different aspect are marked in bold. This also demonstrates the efficiency achieved by adding manual filtering of the dictionary after generated using SeededLDA.

Unbalanced Data The figure 4.3 and 4.4 shows the amount and proportion of labeled data in the dataset using Vietnamese E-commerce Dataset [6]. It can clearly observed that the number of negative labels in is much less than that of positive labels. Lacking number of negative labels has bad effect on the model's label recognition efficiency, which is shown in the data tables 4.5. Some notable labels include:

- *authenticity*: When compared to the other labels, this pair of labels has the greatest negative/positive difference. This also has an impact on the low results of these two aspects when categorizing them as negative labeled items. The reason can come from the data as well as the seed set. Due to the lack of words that accurately refer to the aspect of *authenticity*, there are not many words to accurately refer to this aspect, the seed set is not diverse, and the dictionary tends to become confused with other aspects after each generation.

- *service, accessories*: These label have the amount not too large compared to other aspects of associated data labeled, but is relatively balanced between positive and negative. This is also part of the reason the data obtained with these aspect are relatively good.

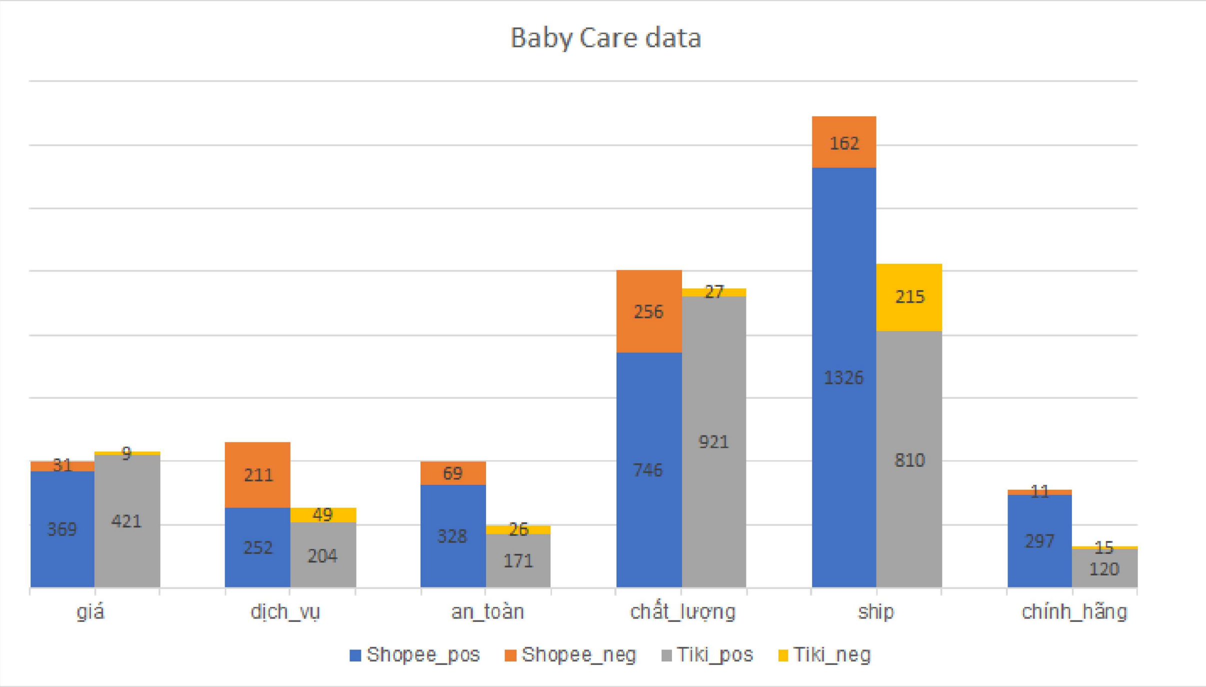


Figure 4.3: Mother & Baby data statistics

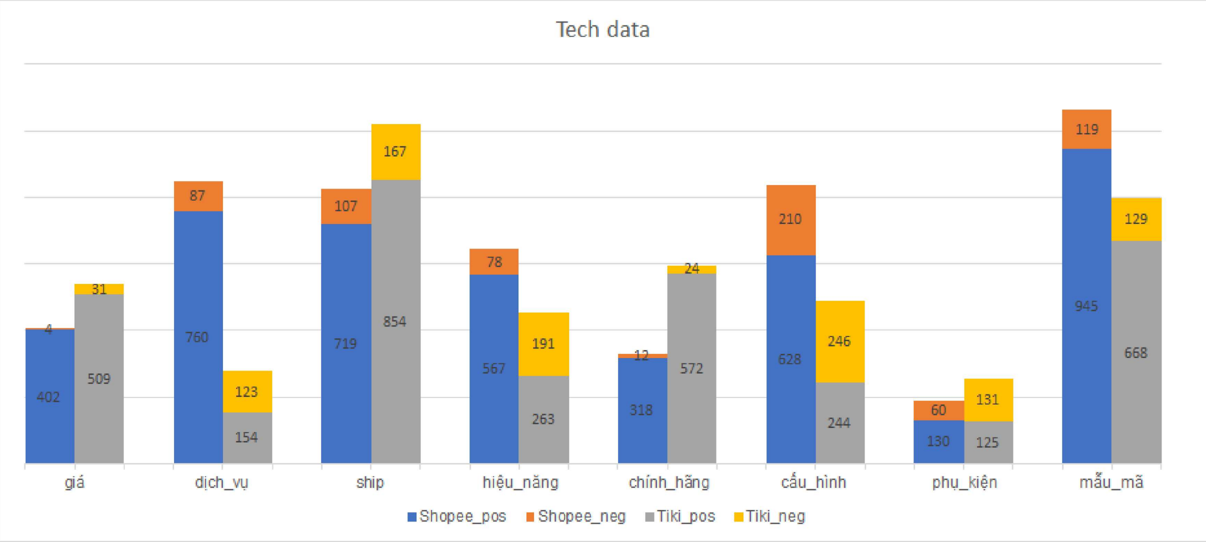


Figure 4.4: Technology data statistics

Conclusions

With the current status and high demand in deploying automated systems to understand the emotional state of users of e-commerce platforms through evaluation comments, this thesis has deployed a model to solve ABSA problem. The thesis applied model based on SeededLDA, a semi-supervised improvement version of the LDA generation model to flexibly use user knowledge through predefined seed words to automatically generate dictionary features; combined with Chi-Square score calculation and key phrase identification using MWLA, which achieves the highest F1 of 99.3% and 90% in terms of “service" aspect for positive and negative label, as well as a 13.5% improvement in macro-f1 compared to the baseline model.

With the existing foundations and the remaining weaknesses, some future directions to overcome and improve the model are as follows:

- Use other data sets to continue making objective comparisons and assessments.
- Fix so that the model can better respond to unbalanced data cases.
- Developing and tuning SeededLDA to reduce the need to manually filter the dictionary while still ensuring efficiency.

List of Publications

[Pub 1] **Le-Minh, Binh**, Thi-Phuong Le, Khanh-Hung Tran, Khanh-Huyen Bui, Hoang-Quynh Le, Duy-Cat Can, Hung Nguyen Chung Thanh, and Mai-Vu Tran. "Aspect-Based Sentiment Analysis Using Mini-Window Locating Attention for Vietnamese E-commerce Reviews." In 2021 13th International Conference on Knowledge and Systems Engineering (KSE), pp. 1-4. IEEE, 2021.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[2] Z. Chen and B. Liu, “Mining topics in documents: standing on the shoulders of big data,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 1116–1125.

[3] E. Fix and J. L. Hodges, “Discriminatory analysis. nonparametric discrimination: Consistency properties,” *International Statistical Review/Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.

[4] J. Jagarlamudi, H. Daumé III, and R. Udupa, “Incorporating lexical priors into topic models,” in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 204–213.

[5] P. Kotler, G. Armstrong, L. Harris, and H. He, *Principles of Marketing*. Pearson, 2019.

[6] B. Le-Minh, T.-P. Le, K.-H. Tran, K.-H. Bui, H.-Q. Le, D.-C. Can, H. N. C. Thanh, and M.-V. Tran, “Aspect-based sentiment analysis using mini-window locating attention for vietnamese e-commerce reviews,” in *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2021, pp. 1–4.

[7] B. Liu, “Sentiment analysis and opinion mining,” *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] S. Poria, I. Chaturvedi, E. Cambria, and F. Bisio, “Sentic lda: Improving on lda with semantic similarity for aspect-based sentiment analysis,” in *2016 international joint conference on neural networks (IJCNN)*. IEEE, 2016, pp. 4465–4473.

[10] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, “VnCoreNLP: A Vietnamese natural language processing toolkit,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 56–60.

[11] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, “A survey on aspect-based sentiment analysis: Tasks, methods, and challenges,” *arXiv preprint arXiv:2203.01054*, 2022.