Table of Contents

Abstract . . . . . **iii**

Table of Contents . . . . . **iv**

Acronyms . . . . . **vi**

List of Figures . . . . . **viii**

List of Tables . . . . . **ix**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Definition 2

1.3 Related Works 2

1.4 Contributions and Structure of the Report 3

2 Dataset Construction **4**

2.1 Data Collection 4

2.2 Data Annotation 4

2.3 Dataset Analysis 6

3 Methods **9**

3.1 Theoretical Basis 9

3.1.1 K-nearest Neighbors 9

3.1.2 Logistics Regression 10

3.1.3 Support Vector Machine 11

3.1.4 Naive Bayes 11

3.1.5 Decision Tree 12

3.1.6 Random Forest 12

3.1.7 Convolutional Neural Network 12

3.2 Model overview 13

3.3 Preprocessing . . . 14
3.4 Locating . . . 15
3.4.1 Chi-Squared Test for Feature Selection . . . 15
3.4.2 Word-Window Locating . . . 16
3.5 Representation . . . 17
3.5.1 One Hot encoding . . . 17
3.5.2 Chi-Squared . . . 17
3.5.3 Word Embedding using PhoBERT . . . 18

4 Experiments and Results . . . 20
4.1 Experiment Setup . . . 20
4.1.1 Experimental Data . . . 20
4.1.2 Baseline Methods . . . 20
4.1.3 Evaluation Metrics . . . 20
4.2 Experiment Result and Analysis . . . 22
4.2.1 Classic Models with One-Hot Representation Method. . . . 22
4.2.2 Logistics Regression with Different Representation Methods . . . 22
4.2.3 Chi-Squared with/without Word-Window Locating compared with CNN + PhoBERT . . . 23
4.2.4 WWL + CS detailed statistics in all domains . . . 24
4.2.5 Error Analysis . . . 24

5 Conclusions . . . 25

Conclusions . . . 25

References . . . 27

Chapter 1

Introduction

1.1 Motivation

In recent years, an increasing number of sentiment data or reviews are produced together with the explosion of product discussion on the Internet, especially on E-commerce platforms. Not only play a significant role in consumer's purchase-decision process, these reviews also important to businesses as they could monitor brand and product sentiment in customer feedback as well as understand customer needs. However, tremendous amount of data make manual approach time-consuming and costly and therefore impractical, which truly create space for automated methods.

Sentiment Analysis (SA) is a natural language process technique with a view to extract the sentimental status of opinions as positive, negative or neutral. This topic has been studied very early in the world in general (Turney, 2002 [11]) and Vietnam in particular (Kieu and Pham, 2010 [3]). However, the main problem in SA is to determine sentence-level sentiment, which is inadequate for further analysis needed. For example, in the Vietnamese sentence *"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") the mentioned aspects are *"chất lượng"* or quality and *"giao hàng"* or delivery. Each opinion can refer to more than one aspect, as well as specific aspect is usually implied rather than general or entity one (which can be solved by using normal SA solutions). To address this problem, we need deeper analysis in terms of aspect-level sentiment called aspect-based sentiment analysis.

1.2 Problem Definition

Aspect-Based Sentiment Analysis (ABSA) is a sub-field of sentiment analysis, which allows us to deeply understand and determine sentiment in terms of different aspects of the topic (Thin et al., 2018 [12]). An ABSA system receives textual data (e.g., reviews or comments on shopping platforms) about specific entity (e.g., baby care products like diapers). The system must be able to detect the mainly discussed aspects of the entity (e.g., "`quality`", "`delivery`"), together with the sentiment of each aspects, or how positive/negative the opinions are. So basically the ABSA problem can be divided to two sub-task: aspect detection and sentiment polarity detection. The first sub-task attempts to determine all aspects from opinion, meanwhile the second sub-task aims to decide which sentiment polarity each aspect is. Considering above example "Soft, thin, good quality diapers, absorb a lot but delivery is quite slow", the system must be able to determine all the aspect-sentiment tuple: {`quality`, `positive`} and {`delivery`, `negative`}.

1.3 Related Works

Due to broad applications in giving necessary details on different aspects of the sentence or document, ABSA has been extensively researched in various languages. ABSA was first researched and introduced by (Hu and Liu, 2004 [1]) in which they only aim to determine product features/aspects that the reviewers have commented on. In the past few years, neural network-based systems have became trending adaptation for ABSA problem as these methods can be trained end-to-end and automatically learn important features (Jiang et al., 2019 [2]). (Wei and Tao, 2018 [14]) developed Gated Convolutional Network, a model based on convolutional neural networks and gating mechanisms. (Yukun et al., 2018 [4]) proposed Sentic LSTM, an extension of long-short term memory (LSTM) network. (Nguyen and Shirai, 2015 [7]) introduced recursive neural network approach to make the representation of the target aspect richer by using syntactic information. In Vietnamese, various methods were proposed in the recent years to address ABSA problem with Vietnamese textual data such as BRNN-CRF architecture (Mai and Le, 2018 [5]), SVM-based model (Thin et al., 2018 [12]), Semantic Relation Analysis (Tran and Phan, 2018 [10]), semi-supervised learning (Nguyen-Nhat et al., 2019 [8]), etc.

1.4 Contributions and Structure of the Report

The main contribution of this work are:

(i) We build a data set for training and testing the sentiment classification model. Implementation steps include surveying, collecting, normalizing, annotating, calibrating, and analyzing data.

(ii) We propose a multi-label classification model for the second sub-task of ABSA. Word window is used to extract information of specific aspects and various data representation methods to extract features for classifiers. Afterwards, we apply classic models concurrently with Deep Learning adaptation to find out best approach for our situation.

The remainders of the paper is organized as follows. Chapter 2 describes the process of dataset construction: collection, annotation and analysis. The research directions of the ABSA problem relevant to dedicated domain (Technology and Mother & Baby in detail) is represented in chapter 3. Chapter 4 describes the experiments that we illustrates the statistics of all methods considered in each of our model's components and finally, chapter 5 concludes the topic with our future works.

Chapter 2

Dataset Construction

2.1 Data Collection

To serve our dedicated approach, we have to handle textual data that are reviews on E-commerce platforms. We choose Scrapy[1] to crawl the data from online shopping websites including `http://www.tiki.vn` and `http://www.shopee.vn`. They are two of Vietnam's most common and trusted e-commerce platforms. We focus on technology and mother & baby domains since they are the most interested domain in Vietnamese e-commerce. These raw data are then used as input for Data Annotation step that we will describe in Section 2.2.

2.2 Data Annotation

We used Docanno as a tool for data annotation. First of all, we overview the data set collected from crawling process to figure out aspects that were frequently mentioned on users' reviews.

After discussion, we extracted aspect terms in the sentences and labeled the sentiment polarities with respect to the aspect terms using Docanno. Table 1 illustrates how we handle each aspect. For Technology domain, we predefined eight coarse aspect categories: price, service, delivery, performance, hardware, authenticity, accessories, design. For Mother & Baby, these are: price, service, delivery, safety, quality and authenticity. In each aspect, it is considered negative when there is at least a complaint regarding that aspect, otherwise it is considered negative.

[1] `https://scrapy.org/`

Domain	*Aspect*	*Description*
Technology	Price	Cut/ reduce/ slash/ low price considered positive while increase/ put up/ raise/ high price is considered negative.
	Service	Nice, fast, efficient, enthusiastic support is considered positive while no reply, irresponsibility or carelessly packing is considered negative.
	Delivery	The quality, speed and cost of shipping process. If it is fast, carefully shipped and the cost is low, it is considered positive, otherwise is negative.
	Performance	Fast processing speed is considered positive while lag or latency in the middle of an app or apps shut down unexpectedly is considered negative.
	Hardware	The quality of the hardware of the device including display, chip, battery, cameras, storage, RAM, etc.
	Authenticity	Genuinely produced product is considered positive while fake, imitation is consider negative.
	Accessories	Fully provided, good quality accessories are considered positive; low quality or missed are considered negative.
	Appearance	Product with nice design, nice color, luxury, etc. are considered positive, product with scratches, awful design is consider negative.
Mother & Baby	Price	Cut/ reduce/ slash/ low price considered positive while increase/ put up/ raise/ high price is considered negative.
	Service	Nice, fast, efficient, enthusiastic support is considered positive while no reply, irresponsibility or carelessly packing is considered negative.
	Delivery	The quality, speed and cost of shipping process. If it is fast, carefully shipped and the cost is low, it is considered positive, otherwise negative.
	Safety	Product having obvious expiration date and has not expired, safe to use is considered positive, while product that is expired or causes allergy, rashes is considered negative.
	Quality	Customer experience on the product: the softness, absorbency of diapers, the taste and smell of milk, etc.
	Authenticity	Genuinely produced product is considered positive while fake, imitation is consider negative.

Table 2.1: Aspect description

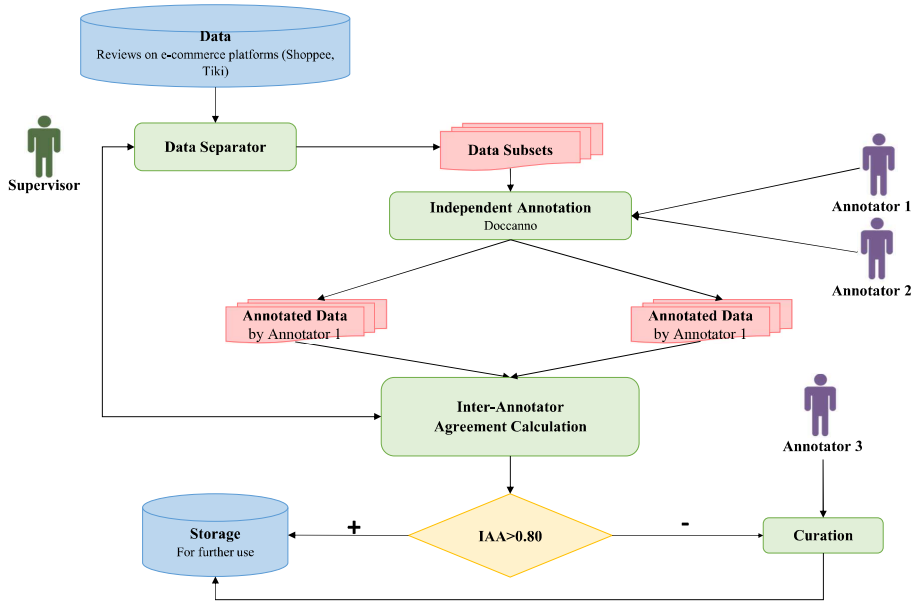

Figure 2.1: Data Annotation process

Data then separated in subsets, each of them was then evaluated and labeled by 2 annotators independently. Next, we measured the agreement between two annotators, if the result is too low (Inter-Annotator Agreement or IAA < 0.8), manually reviewing – process will be applied. The labeled data then stored as input for Pre-processing step.

2.3 Dataset Analysis

The statistics of Vietnamese E-commerce dataset (VECD) including Shopee and Tiki in two different domains: Technology and Mother & Baby are reported in figure 1 to 4. VECD consists of 3016 instances of Shopee Mother & Baby, 2986 instances of Tiki , 3002 instances of Shopee Technology and 3236 instances of Tiki Technology, making up 12240 instances in total.

In Mother & Baby domain, Shipping and Quality appear to be the most concerned aspects. Shopee has 1541/3216 comments on Shipping and 773/3216 comments on Quality, while in Tiki they are 972/2986 and 1177/2986 respectively. Authenticity only occa-

	Mother & Baby		Technology	
	Shopee	***Tiki***	***Shopee***	***Tiki***
Total aspect	6	6	8	8
Aspect count per sentence	1.753234	1.497347	2.087627	1.746551

Table 2.2: The average number of aspects mentioned

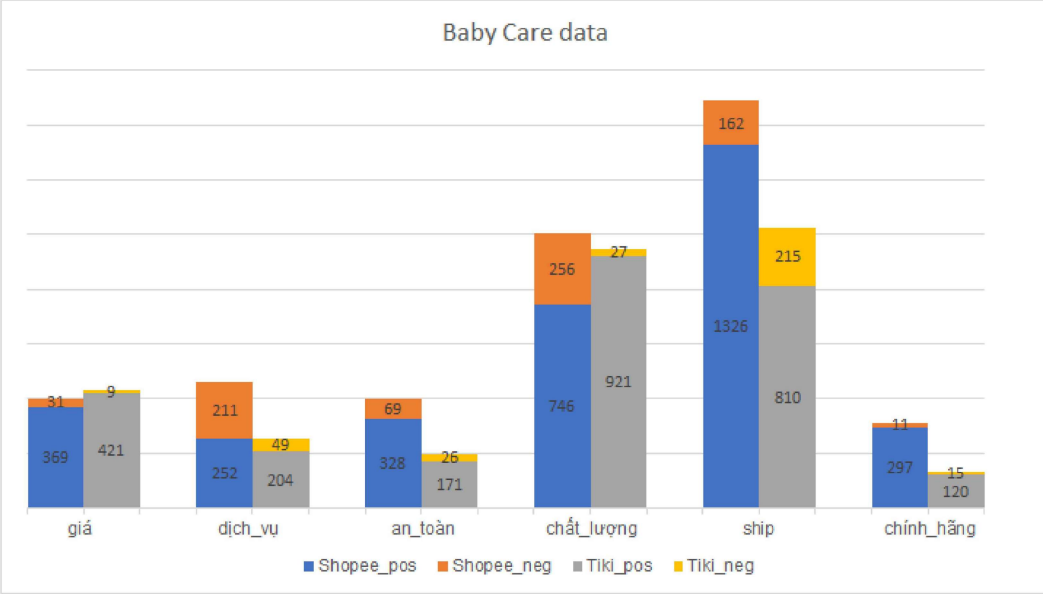

Figure 2.2: Mother & Baby data statistics

sionally mentioned in Tiki's comments, just in 131 cases while the others are mentioned in over 200 comments, in both platforms, including Authenticity mentioned in Shopee.

Similarly, in Technology domain, Appearance is frequently mentioned in Shopee while it is Shipping in Tiki. More specifically 1064 out of 3002 comments on Shopee have information relating to products' appearance, compared to 1021 out of 3236 comments on Tiki having information about Shipping. Price and Authenticity are the most imbalanced aspects, with only 4 negative comments on Price and 12 negative comments on Authenticity in Shopee while in Tiki the figures are 31 and 24, respectively. In Tiki, although the number of comments on Service, Hardware, Performance and Accessories is comparatively low, the data on these aspects are relatively balanced, with the differ-

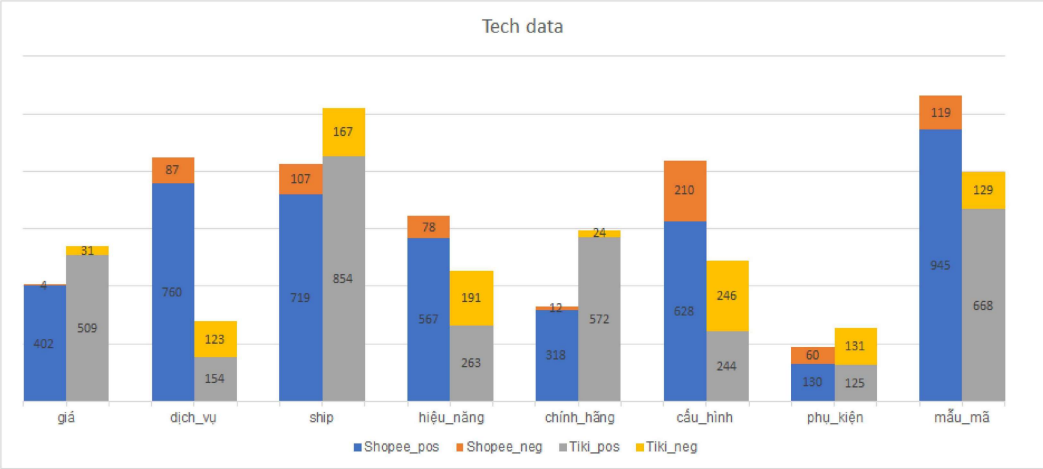

Figure 2.3: Technology data statistics

ence below 16%.

Chapter 3

Methods

In this chapter, we firstly present the methodology for our approaches, then describe our aspect-oriented model in detail, begin with an overview of the whole architecture, followed by the explanation of how each of the components work; lastly, we introduce the parameter system used in the model.

3.1 Theoretical Basis

3.1.1 K-nearest Neighbors

KNN is one of the most straightforward supervised algorithms, but effective in dealing some problems, in both classification and regression. KNN does not study anything from training data (therefore it's considered as a lazy learning method)

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.

Despite lightning performance in small dataset, as the data size enlarges, KNN is not beneficial for both memory and time. The algorithm must remember all the training

data for label calculation, so it will become extensively slow as size of dataset grows.

3.1.2 Logistics Regression

LR, in its basic form, uses a logistic function (e.g., sigmoid, tanh) to model a binary dependent variable. The prediction score of LR 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 LR 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.

LR is one of the easiest ML algorithms as it is easy to implement, interpret, and very efficient to train. Traning a model with LR doesn't need high computation effort. LR 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 (SGD).

But LR 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. LR 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.

3.1.3 Support Vector Machine

A SVM 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, SVM maximizes the margins from both class.

With nonlinear data, additional dimensions will be required. SVM can classify vectors in multidimensional space.

3.1.4 Naive Bayes

NB 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.

NB 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:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \tag{3.1}$$

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

3.1.5 Decision Tree

A decision tree is used to represent decisions and decision-making process. The tree can be explained as the leaves, which is final outcome, and the decision node, which is the place data splits.

For classification purpose, the outcomes (or leaves) are corresponding to features, which is selected throughout the tree. Begin with the root, or the starting node, we find out the best decision (e.g., best attribute) by applying selection measures (Information Gain, Gini Index, etc.) in the dataset, then divide dataset into subsets and recursively perform the process until the nodes can not be further divided. The leaf nodes now is the final outcome, and the total process make out the model.

3.1.6 Random Forest

RF is the combination of multiple DT that opearate as an ensemble, which results in more reliable and stable prediction. Each DT finds out a class prediction, and the most chosen becomes the total model's prediction. As those DTs are relatively independent, RF is protected from individual DT's error (unless all the DTs are wrong in the same way).

RF works well with categorical variables. Missing or imbalanced values can be handled easily. Noise or new data point also not a problem as it can only affect some of DTs but not to entire forest.

3.1.7 Convolutional Neural Network

CNN is a popular Deep Learning model. A typical CNN consists of a set of basic layers including: convolution layer with kernels, pooling layer, fully connected layer, which are linked together in a certain order.

Convolution is the first class to extract the features from the input sentence. Convolution maintains relationship between words by exploring sentence feature by using small segments of input data. It is a mathematical operation that takes two inputs such as a matrix of words and filter of kernels.

The maxpooling layer will reduce the number of parameters when the word count is too large. Spatial aggregation is also known as sub-sampling, which reduces the dimensionals of each map but retains important information.

The last layer flattens its matrix into vector and brings it to a fully connected layer like a neural network, which combines the features together to create a model.

Finally, we have a activation function like softmax or sigmoid to to classify with probability value from 0 to 1.

3.2 Model overview

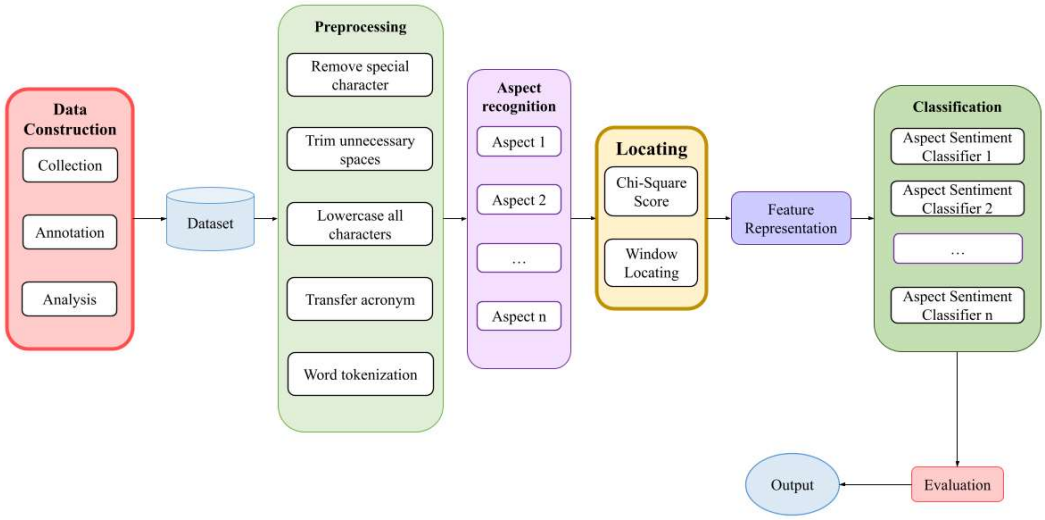

Figure 3.1: The aspect and sentiment analysis pipeline

The main objective of our model is to address the second sub-task of the ABSA problem as it must be able to perform the aspect polarity task. In this case, the system should identify which of two polarity labels - positive or negative could be assigned with the corresponding aspect. Figure 3.1 shows the position of our problem in the opinion mining overall architecture. In which aspect recognition is considered as the preceding problem of sentiment classification and is outside the scope of this study. In this report, we assume that aspect recognition is completed before.

Our aspect-oriented sentiment analysis model consists of four main phases, illustrated in Figure 3.2: reprocessing, locating, representation and classification. The first phase helps cleaning and preparing data for classification, the second one applies word-window method to find out the words which has a high ability to decide which sentimental status the aspects are. Following to that, we tested different ways to represent words with a view to extract meaningful features from data. Finally, the last phase is

our application of multiple classifiers, including both classic ones like SVM, Random Forest, etc. and modern one as Deep Learning, which will be all described within this section.

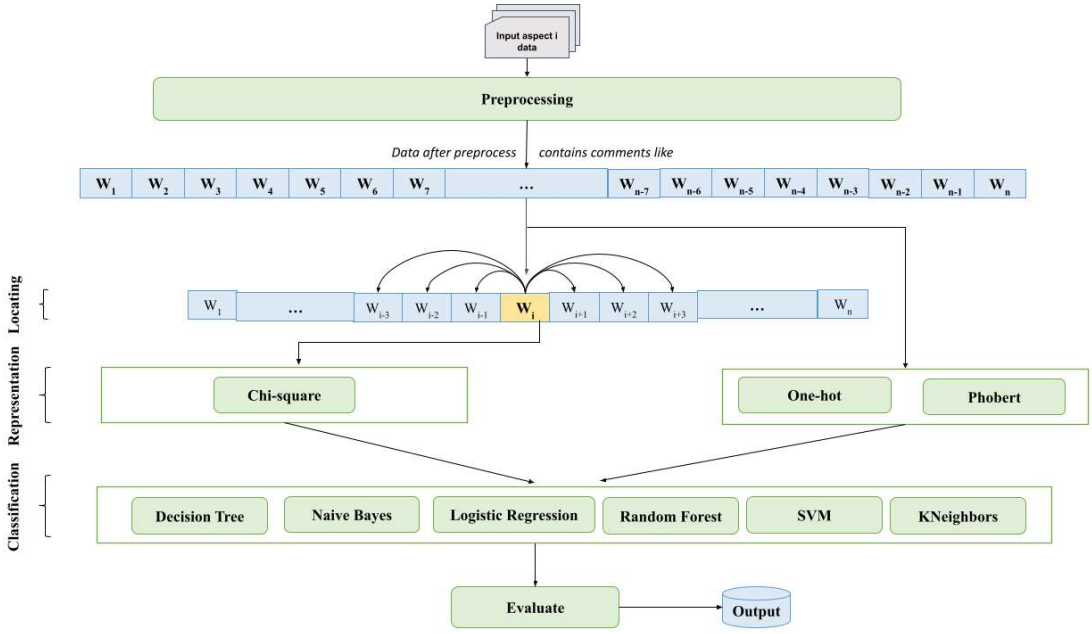

Figure 3.2: Implementation steps for a single aspect

3.3 Preprocessing

Preprocessing is one of the key steps in every natural language processing problem as it transforms data into usable one which machine can easily interprete. Since the characteristics of the input data are raw text collected from websites, the input text (from the dataset contains the specific aspect that we have gathered and processed in Chapter 2 above) can have noise which can harmful to machine learning performance such as special characters, spelling mistakes, spacing errors, etc. To standardize the data and reduce the amount of noise, preprocessing process is applied according to the following steps:

- **Step 1:** Special characters (including punctuations) were completely removed.
- **Step 2:** Trim unnecessary spaces made by spacing errors.
- **Step 3:** Lowercase all characters.
- **Step 4:** Transfer acronyms to the full form of them and translated from context.

Input	**Bỉm mềm , chất lượng tốt , hút dc nhiều nhưng giao hàng khá chậm.**
Step 1	Bỉm mềm chất lượng tốt hút dc nhiều nhưng giao hàng khá chậm
Step 2	Bỉm mềm chất lượng tốt hút dc nhiều nhưng giao hàng khá chậm
Step 3	bỉm mềm chất lượng tốt hút dc nhiều nhưng giao hàng khá chậm
Step 4	bỉm mềm chất lượng tốt hút được nhiều nhưng giao hàng khá chậm
Step 5	**bỉm mềm chất_lượng tốt hút nhiều giao_hàng chậm**

Table 3.1: Step-by-step preprocessing illustration

- **Step 5:** Word tokenization. Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units called tokens, such as individual words or terms. By tokenization, the meaning of the text can be interpret easier by some analysing methods such as count the number of words appeared, the frequency of the word, and so on.

  VnCoreNLP (Vu et al., 2018 [13]) is used for all pre-processing steps.

3.4 Locating

3.4.1 Chi-Squared Test for Feature Selection

In statistics, the Chi-squared test is used to determine whether two categorical variables independent or related. In feature selection, the two variables are the observations of the feature and the occurrence of the class. The outcome of the test is a test statistic that has a chi-squared distribution and can be clarified or fail to reject the assumption or null hypothesis H_0 that the observed and expected frequencies are equal.

Given a document D, we estimate the (χ^2) value and rank them by their score:

$$\chi^2 = \sum_{t=1}\sum_{c=1} \frac{(O_{t,c} - E_{t,c})^2}{E_{t,c}} = N \sum_{t,c} p_t p_c \left(\frac{(O_{t,c}/N) - p_t p_c}{p_t p_c} \right)^2$$

where

χ^2 = Pearson's cumulative test statistic, which asymptotically approaches a χ^2 distribution.

$O_{t,c}$ = the number of observations of type t in class c.

N = total number of observations.

$E_{t,c} = Np_{t,c}$ = the expected (theoretical) count of type t in class c, asserted by the null hypothesis that the fraction of type t in class c in the population is $p_{t,c}$

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 we should select the feature for classification. In other words, using this method remove the feature that are most likely autonomous of class and consequently unessential for classification.

We use Scikit-learn (Pedregosa et al., 2011 [9]) as it provide multiple feature selection methods, including Chi-Squared Test. Scikit-learn gives a *SelectKBest* class that can be used with various statistical tests. It will rank the features with the statistical test that we've determined (Chi-Squared Test in detail) and select the top k performing ones (implying that these terms is viewed as more relevant to the task than the others). These top performing features will be used for locating and one of the representation method that we will described below.

3.4.2 Word-Window Locating

We use 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 sentence as a set of word $W = \{w_0, w_1, w_2, ..., w_n\}$ and a set of class $C = \{c_0, c_1, c_2, ..., c_m\}$, we simply define score $s_{i,j}$ as χ^2 score of the word w_i in class c_j. For each aspect c_j , we choose a threshold s_{c_j}, and ignore all the words which have lower score than that threshold. If the sentence has no word with score higher than threshold, the word with highest score will be selected. Based on calculation above, we take out 3 words before and after the high-score word w_i. This combination, or *window*, can be illustrated as: $\{w_{i-3}, w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, w_{i+3}\}$

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

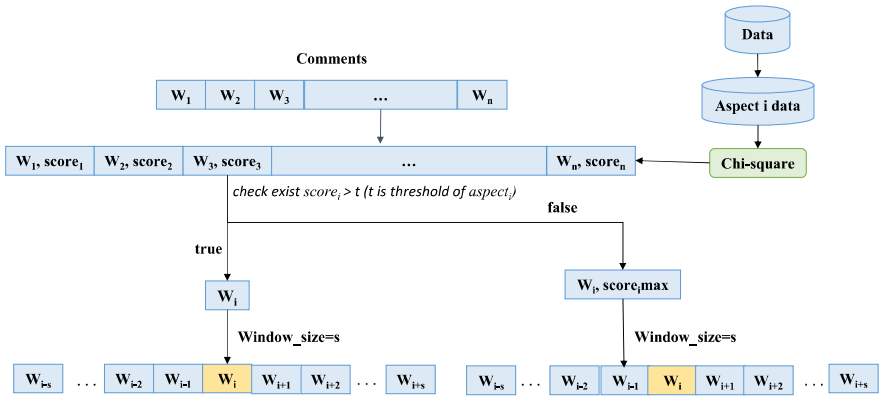

Figure 3.3: Word-Window Locating process

3.5 Representation

3.5.1 One Hot encoding

Every word which are part of the text data are written in the form of vectors, constituting only of 1 and 0 (1 if word in vocabulary else 0). Although it does not highlight the importance of words in sentiment classification, but it has yielded quite surprising results. Further information will be provided in Section 4.2.

3.5.2 Chi-Squared

We use word-level χ^2 as calculated above to weight the words in the vocabulary, and use that vocabulary to represent data. Then we proceed to filter each aspect's vocabulary manually to reduce dimensions used for classfiers. The results using this data representation method shown in Section 4.2.

3.5.3 Word Embedding using PhoBERT

BERT BERT (Bidirectional Encoder Representations from Transformers) is understood as a pre-trained model, which learns vectors that represent two dimensions of words while models such as Word2vec, fastText find a vector that represents each word based on a large set of materials, so it does not represent the diversity of context. BERT has succeeded in improving recent work in finding representatives of words in digital spaces (spaces that computers can understand) through its context.

Transformer is an attention mechanism that learns the correlation between words (or part of a word) in a text. Transformer consists of two main parts: Encoder and Decoder, encoder that reads input data and decoder makes predictions. In this sitution, BERT uses only Encoder.

BERT is used to support other problems in the field of natural language processing such as emotional classification, amantic analysis, spam filtering, news classification, etc. Our proposed work use BERT for the problem of sentment classification.

PhoBERT Pre-trained PhoBERT models are the state-of-the-art language models for Vietnamese (Dat and Anh, 2020 [6]). PhoBERT uses VnCoreNLP to separate words for input data before passing encoder.

PhoBERT output consists of 2 layers: the first one is the last hidden layer created by token embedding, sentences embedding, transformer positional embedding into a 3D vector, while the second is the layer that has been pooling into a 2D vector.

In each string of words after using PhoBERT, we have two ways of expressing the word and sentence level:

- Represented by word level: each word will be represented as a 768-dimensional vector, so in total a $n*768$ matrix will represent a sentence.
- Represented by sentence level: the whole sentence is also represented by a 768-dimensional vector.

Typically, a sentence after being represented by the matrix bar will be spread through the convolution layer + nonlinear layer first (RELU), after which the calculated values will spread through the pooling layer, multiple kernels.

Specifically, the word vectors from w_i to w_{i+j-1} as $[w_i, w_{i+1}, w_{i+2}, ..., w_{i+j-1}]$ will

be combine to represent one feature. Afterwards, a filter is applied to the specific k-word vector (in the proposed work we use k = 2, 3 and 4) to captured most useful features for sentiment detection.

Feature c_i is generated from a nonlinear activation function (hyperbolic tangent or ReLU)

$$c_i = f(F(w_{i,i+k} + b) \tag{3.2}$$

where b is a bias, a vector performs as a additional function to the input.

We will get 1 characteristic matrix for a specific feature: $c = [c_1, c_2, c_3, ..., c_h]$ Then, maxpooling (take out the maximum characteristic c_i among the matrix) is applied to choose the most important features. The reason for choosing the highest c is because the input will inevitably map to a fixed output size so that it is fully connected, the input size is reduced but still maintains important properties.

The convolution filer number is certain with different width and slips across the entire matrix to get all the properties.

Chapter 4

Experiments and Results

4.1 Experiment Setup

4.1.1 Experimental Data

In preparation to evaluate the effectiveness of our model, we proposed experiments on the Baby Care dataset collected from Shopee and Tiki, which has been described previously.

4.1.2 Baseline Methods

We compare various combinations of classifiers and different representation methods to figure out which one is most effective for our situation. The evaluation process is:

(1) Classic models: Logistics Regression, Random Forest, Decision Tree, Naive Bayes, K-nearest Neighbors and Support Vector Machine combine with different representation methods: One-Hot, Chi-Squared, PhoBERT.

(2) Logistics Regression with different representation methods.

(3) General comparison with/without WWL to find out the best model.

4.1.3 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.

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

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.

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

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.

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

Micro-average and Macro-average Performance Micro Average and 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, whereas a

	Macro-average	***Micro-average***
P	$macro_P = \frac{\sum_i P_i}{N}$	$micro_P = \frac{\sum_j TP_j}{\sum_j TP_j + \sum_j FP_j}$
R	$macro_R = \frac{\sum_i R_i}{N}$	$micro_R = \frac{\sum_j TP_j}{\sum_j TP_j + \sum_j FN_j}$
F1	$macro_{F_1} = \frac{\sum_i F_{1i}}{N}$	$macro_{F_1} = 2\frac{micro_P * micro_R}{micro_P + micro_R}$

Table 4.1: Micro-average and Macro-average Performance

micro-average will aggregate the contributions of all classes to compute the average metric. Micro- and macro-average for Precision, Recall and F1-score is calculated by the formulas as shown in table 4.1.

4.2 Experiment Result and Analysis

In context of facing imbalance data (positive outnumbered negative) as analysed in Section 2.3, we appreciate the results of class negative in particular. In this section, we will use Micro/Macro-average metrics calculated for negative class as the measure of evaluation.

4.2.1 Classic Models with One-Hot Representation Method.

With the OH data representation method, we use the Mother & Baby data collected from Tiki as the input data for traditional machine learning models. Applying this simple method surprisingly brings fairly good results for such unbalanced data (compared to the methods below) but the negative value of some aspects is not yet assigned.

Based on the results of applying models, we find that the model using the LR method has the most outstanding results. We will use a combination of this method and the data representation methods below to compare results.

Evaluation results are illustrated in table 4.2.

	SVM	***DT***	***KN***	***LR***	***NB***	***RF***
micro-p	0,587	0,622	0,541	0,810	0,205	0,818
micro-r	0,228	0,599	0,370	0,525	0,746	0,500
micro-f1	0,329	0,610	0,440	**0,637**	0,322	0,621
macro-p	0,330	0,422	0,389	0,541	0,183	0,548
macro-r	0,159	0,397	0,258	0,351	0,641	0,329
macro-f1	0,176	0,406	0,299	0,405	0,277	0,395

Table 4.2: Micro/Macro-average for class negative of OH + traditional models

4.2.2 Logistics Regression with Different Representation Methods

We use the model using the LR method for comparison since it is the model with the best results as calculated above. Mother & Baby data collected from Tiki are used to

evaluated results among three different representation methods: PhoBERT, CS and OH. OH and PhoBERT showed relatively good results, but in general CS has the best performance. Table 4.3 illustrates micro- and macro-average performance for class negative of LR with multiple representation methods.

	PhoBERT	***CS***	***OH***
micro-p	0,764	0,704	0,810
micro-r	0,540	0,660	0,525
micro-f1	0,633	**0,682**	0,637
macro-p	0,506	0,560	0,541
macro-r	0,368	0,532	0,351
macro-f1	0,423	**0,545**	0,405

Table 4.3: Micro/Macro-average for class negative of LR with representation methods

4.2.3 Chi-Squared with/without Word-Window Locating compared with CNN + PhoBERT

We continue to use the model using LR method to compare between CS, CS + WWL and CNN + PhoBERT.

The results indicate that CS + WWL outperformed CS alone in every figures. In comparison of CS + WWL and CNN + PhoBERT, it is indicated that micro-F1 of CNN + PhoBERT is slightly better with a percentage of 4% but CS + WWL surpassed CNN + PhoBERT with a significant percentage of 16%.

Evaluation proves effective use of the WWL method.

	CS + WWL	***CS***	***CNN + PhoBERT***
micro-p	0,738	0,704	0,801
micro-r	0,728	0,660	0,747
micro-f1	0,733	0,682	**0,773**
macro-p	0,781	0,560	0,512
macro-r	0,632	0,532	0,518
macro-f1	**0,674**	0,545	0,515

Table 4.4: Micro/Macro-average of CS with/without WWL and CNN + PhoBERT

Table 4.4 shows micro- and macro-average performance of CS with/without WWL and CNN + PhoBERT.

4.2.4 WWL + CS detailed statistics in all domains

Given the proportional difference between the positive and negative classes of different datasets in the domain and the data source in the item (item name), we find that the model uses WWL with the CS method of representing the level data showed quite stable results. The statistical results are described in table 4.5.

	mb_tiki	***mb_shopee***	***tech_tiki***	***tech_shopee***
micro-p	0,738	0,615	0,707	0,709
micro-r	0,728	0,635	0,621	0,629
micro-f1	0,733	0,625	0,661	0,667
macro-p	0,781	0,541	0,582	0,593
macro-r	0,632	0,462	0,510	0,572
macro-f1	0,674	0,482	0,540	0,545

Table 4.5: WWL + CS detailed statistics for class negative in all domains

4.2.5 Error Analysis

- Unbalanced data results affect model training leading to low results for the indicated aspect.
- Lack of summarized data.
- Errors occur in the data annotation phase due to manual process.
- The investigation of the data is not in-depth, yield threshold selected for locating not matching the data.

Chapter 5

Conclusions

Contributions

In this article, we proposed multiple approaches for the sentiment polarity detection - a sub-task of ABSA problem. Experiments indicated significant result as WWL combine with classic models has reach the maximum percentage of 95% in Micro- and Macro-Average Performance of F1-score.

Through experiments using the locating method and apply multiple data representation techniques, we can determine that using the WWL method with the Chi-Squared data representation brings very prominent results when combined with traditional machine learning methods. Because when classify the sentiment for multi-label data (i.e., multiple aspects in a data sample) using multiple models, we need to find the words associated with the label and the word representing its sentiment. WWL technique step solves this problem. In addition, in order for the data to focus on important words, we use Chi-Squared score to weight word level in the vocab and use it to represent data. This combination brings the most outstanding advantage to address the problem.

Limitations and Future Work

Selecting threshold for locating words inappropriately may be harmful to the result of the model. Window size (i.e., the number of words around from the center, which equals 3 in the proposed work) affects final result too. This requires manual data revision to be done carefully to choose which one is the best. This is the limitation of our method.

The weak point of proposed work indicates our next path to have in-depth research

in the future:

(i) More data should be collected to enlarge the dataset.

(ii) Handling the data imbalance problem.

(iii) Developing WWL in order to overcome current disadvantage.

References

[1] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168–177.

[2] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A challenge dataset and effective models for aspect-based sentiment analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6281–6286.

[3] B. T. Kieu and S. B. Pham, "Sentiment analysis for vietnamese," in *2010 Second International Conference on Knowledge and Systems Engineering*. IEEE, 2010, pp. 152–157.

[4] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic lstm: a hybrid network for targeted aspect-based sentiment analysis," *Cognitive Computation*, vol. 10, no. 4, pp. 639–650, 2018.

[5] L. Mai and B. Le, "Aspect-based sentiment analysis of vietnamese texts with deep learning," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2018, pp. 149–158.

[6] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1037–1042.

[7] T. H. Nguyen and K. Shirai, "Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2509–2514.

[8] D.-K. Nguyen-Nhat and H.-T. Duong, "One-document training for vietnamese sentiment analysis," in *International Conference on Computational Data and Social Networks*. Springer, 2019, pp. 189–200.

[9] 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.

[10] T. K. Tran and T. T. Phan, "Towards a sentiment analysis model based on semantic relation analysis," *International Journal of Synthetic Emotions (IJSE)*, vol. 9, no. 2, pp. 54–75, 2018.

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

[12] T. Van Dang, V. D. Nguyen, N. Van Kiet, and N. L. T. Ngan, "A transformation method for aspect-based sentiment analysis," *Journal of Computer Science and Cybernetics*, vol. 34, no. 4, pp. 323–333, 2018.

[13] 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. [Online]. Available: https://www.aclweb.org/anthology/N18-5012

[14] W. Xue and T. Li, "Aspect based sentiment analysis with gated convolutional networks," *arXiv preprint arXiv:1805.07043*, 2018.