

**VIETNAMESE SENTIMENT ANALYSIS FOR ONLINE
BOOKING HOTEL REVIEW BASED ON TERM FEATURE
SELECTION AND DEPENDENCY TREE**

BY

TRAN SY BANG

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF MASTER OF
ENGINEERING (INFORMATION AND COMMUNICATION
TECHNOLOGY FOR EMBEDDED SYSTEMS)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2016**

**VIETNAMESE SENTIMENT ANALYSIS FOR ONLINE
BOOKING HOTEL REVIEW BASED ON TERM FEATURE
SELECTION AND DEPENDENCY TREE**

BY

TRAN SY BANG



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF MASTER OF
ENGINEERING (INFORMATION AND COMMUNICATION
TECHNOLOGY FOR EMBEDDED SYSTEMS)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2016**

VIETNAMESE SENTIMENT ANALYSIS FOR ONLINE BOOKING HOTEL
REVIEW BASED ON TERM FEATURE SELECTION AND DEPENDENCY TREE

A Thesis Presented

By

TRAN SY BANG

Submitted to

Sirindhorn International Institute of Technology

Thammasat University

In partial fulfillment of the requirements for the degree of
MASTER OF ENGINEERING (INFORMATION AND COMMUNICATION
TECHNOLOGY FOR EMBEDDED SYSTEMS)

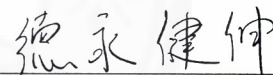
Approved as to style and content by

Advisor and Chairperson of Thesis Committee



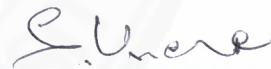
(Dr. Virach Sornlertlamvanich)

Co-Advisor



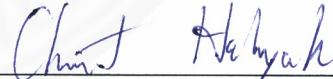
(Prof. Dr. Tokunaga Takenobu)

Committee Member and
Chairperson of Examination Committee



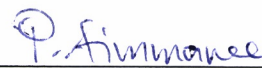
(Dr. Virach Sornlertlamvanich)

Committee Member



(Dr. Choochart Haruechaiyasak)

Committee Member



(Asst. Prof. Dr. Pakinee Aimmanee)

17 MARCH 2017

Acknowledgements

First and foremost, I would like to convey my deepest gratitude to my research advisor Dr. Virach Sonlernlamvanich (SIIT, Thammasat University) for his endless support and advice. I also want to give sincere thanks for Prof. Tokunaga Takenobu (Tokyo Institute of Technology) who was staying in far distance but continuously giving precious feedbacks for my thesis. My thesis would not be completed without initiative idea from Dr. Choochart Haruechaiyasak (NSTDA, Thailand). Last but not least, my presentation would not be successful without correction and encouragement from Dr. Pakinee Aimmanee (SIIT, Thammasat University). My life during my master degree at SIIT would be struggling without living sponsorship from Dr. Suyama (Tokyo Institute of Technology). I am very thankful for Tokyo Tech program which contributed for the scholarship during two academic years of studying.

I would like to thank for KICSS 2015 and EJC 2016 for publishing my work in two international conferences. These papers are very importance for me to fulfill graduation requirement of Master Degree in Information and Communication Technology for Embedded Systems, Tokyo Tech program. I also take this opportunity to express my gratefulness to staffs from SIIT for assisting me during the graduate study period and giving me a wonderful time of being an examination proctor.

I would like to extend my appreciation for friends in Asian Institute of Technology for helping me to find accommodation in beginning of my master program. Special thanks to Mr. Faisal Ghaffar Khan to be my best buddy so that I did not have a feeling of isolation in the classroom. Without him, many projects and assignments were not successfully compiled.

Finally, I would like to present warmly words to my girlfriend Quyen for being with me during this nomadic stage. Last but not least, I would like to dedicate this moment to my parent and my brother for their endless love and support.

Abstract

VIETNAMESE SENTIMENT ANALYSIS FOR ONLINE HOTEL BOOKING BASED ON TERM FEATURES SELECTION AND DEPENDENCY TREE

by

TRAN SY BANG

Bachelor of Science in Engineering, Asian Institute of Technology, 2014
Master of Engineer, Sirindhorn International Institute of Technology, 2016

This thesis paper presents an unsupervised method to classify document sentence's level into sentiment-oriented categories such as positive and negative. Traditionally, sentiment analysis often classifies sentence based on word features, syllable features, N- Gram features. The sentence in a whole can contain several phrases and words in a meaningful way. However, classification a sentence based on phrases and word can be sometimes incoherent, because they are ungrammatically formed. For solving this kind of limitation, it is important to arrange words and phrase in a semantically strong form. Thus we transform a sentence into dependency tree structure. Dependency tree can hold several subtrees, and each subtree allocates words and syllables in a correct grammatical order. Moreover, a sentence dependency tree structure can mitigate word sense ambiguity or solve the inherent polysemy of words by determining a correct word sense. In our experiment, we provide a detail of used method, and we also present an analysis of our experiment effectiveness.

Keywords: Sentiment analysis, Sentence dependency parsing, subtree opinions, Vietnamese sentiment classification, hotel review classification.

Table of Contents

Chapter	Title	Page
	Signature Page	i
	Acknowledgeme	ii
	Abstract	iii
	Table of Contents	iv
	List of Table	vi
	List of Figures	vii
1	Introduction	1
	1.1 Introduction	1
	1.1.1 Summary	1
	1.2 Background of the study	1
	1.3 Problem statement	2
	1.4 Study objectives	2
	1.5 Scopes of the study	3
	1.6 Thesis outline	3
2	Literature Review	4
	2.1 Sentiment classification by features selection	4
	2.3 Proposed experimental design	15
3	Methodology	16
	3.2 Experimental data set	16
	3.3 Data format for experiment	19
	3.4 Classification techniques	19
	3.4.1 Support Vector Machine (SVM)	21
	3.4.2 K-Nearest Neighbor (KNN)	21
	3.4.3 Naive Bayes	21
	3.4.4 Decision Tree (J48)	22
	3.5 Feature Selection Techniques	22

3.5.1	Information gain	22
3.5.2	χ^2 (CHI)	22
3.6	Method used	23
3.6.1	Baseline methods	23
3.6.2	Machine learning and feature selection techniques	23
3.6.3	System evaluation method	24
3.7	Vietnamese sentiment classification based on dependency parsing	24
3.8	Syntax Theory	26
3.8.1	Syntax and Grammar	26
3.8.2	Dependency structure grammar	26
3.9	New system approach	27
3.10	Projective dependency parsing	27
3.11	Classification with sub opinion relation	30
3.12	Classification with sub-relation	31
3.13	Classification with considering word granularity	32
3.14	System evaluation method	34
3.15	Summary	35
4	Result and Discussion	36
4.1	Results and Analysis for Term Feature Selection	36
4.2	Results and Analysis for Sentence Dependency Parsing	38
4.2.1	Tree and sentence bracketed with Sum-production propagation	39
4.2.2	Tree and sentence bracketed with rewarded and reversed voting	39
4.2.3	Tree and sentence bracketed with Word Sense	39
4.2.4	Overall evaluation	40
4.3	Comparison with other experiments	40
4.4	Summary	43
5	Conclusion and Future Work	44
	References	45

List of Tables

Tables	Page
2.1 Performance of Morgan techniques	5
2.2 The result of experiment	7
2.3 Performance result from Kieu's research	8
2.4 Recognition results on three types of sentences	8
2.5 Classifier and Features used in experiment	9
2.6 The experimental result	9
2.7 The corpus metric of the study	11
2.8 Accuracy of Sentiment Classification	11
2.9 The classification result based on subgraph method	12
2.10 Result from experiment conducted in 13 languages	13
2.11 Error analysis	14
3.1 Format of Vietnamese corpus	19
3.2 The corpus metric	19
3.3 Header and data section of Arff format	20
3.4 Accuracy results of parser	28
3.5 CoNLL format	29
3.6 Corpus volume and extracted features	29
3.7 Sense for đá and hành lý	33
3.8 Confusion matrix table	34
4.1 The result of sentiment classification	36
4.2 Comparative result among used methods	39
4.3 Performance of the system on Positive and Negative classes	41

List of Figures

Figures	Page
2.1 Contextual polarity disambiguation task system description	5
2.2 Comparison between proposed set of features	7
2.3 The sentence dependency tree with extracted sub-trees.	11
2.4 A syntactic annotation dependency graph constructed by Shilpa method	12
2.5 Subgraph features.	13
3.1 System overview	16
3.2 The general evaluation box (Agoda.com, 2016).	17
3.3 The individual comment box (Agoda.com, 2016)	17
3.4 Main tasks in raw text pre-processing.	18
3.5 Structure of data section.	20
3.6 The framework for baseline experiment.	23
3.7 The new system with feature selection.	23
3.8 Original sentence dependency tree	24
3.9 Polarities of Dependency Tree and Sub Tree	25
3.10 Sentence clausal element	26
3.11 New system approach based on dependency parsing	27
3.12 Vietnamese projective dependency tree transformed from Treebank	27
3.13 Bracketed sentence with semantic relation	29
3.14 Subtrees extracted from bracketed sentence	30
3.15 Node features and edge features in dependency tree	31
3.16 Calculating of sub tree polarity	31
3.17 VietWordNet hierarchical structure	33
3.18 Example of expressed sentence	32
4.1 The accuracy of the Decision Tree, Naïve Bayes and SVM	37
4.2 The result with Information Gain feature selection	37
4.3 The result with CHI square selection	38
4.4 Result of Sentiment Analysis for Vietnamese by Duyen	41
4.5 Performance for each class	42
4.6 Comparison with previous experiment	42

Chapter 1

Introduction

1.1 Introduction

1.1.1 Summary

This chapter introduces the trend of online hotel reviews and the problem of Vietnamese sentence level classification, the proposed research, the methodology and the expected output of this research. This research entails a study of the development of text processing tools that employ online retrieval corpus, available machine learning algorithms, and non-projective dependency parser for conducting Vietnamese language processing. This study also contains new research finding of sentence level classification based on term features selection and dependency parsing tree. In addition, the thesis structure will be presented at a flow of study's background, statement of problem, objective and scope of the study.

1.2 Background of the study

Opinion mining (OM) is the basic concept of information retrieval and computational linguistics which is not only concentrated on the theme of the document but also opinions it contained. OM has a large scope of coverage regarding of investigating users' opinions about a typical product or about a social event that is emerging in online forums, to customer services (Andrea, 2016). In recent years, the born of the online platform such as blogs, e-commerce sites generate a tone of digital information. People can freely present their opinions about diversified product specifications. Those feedbacks are the main resource for entrepreneurs and manufacturers to develop and improve the products and to serve their potential customers (Subhabratam, 2012). In fact, customers' reviews classification on hotel quality is another domain that can be taking care by OM.

The process of mining opinions has widely gained the attention of people. We invented many techniques to assist us in handling opinion information and makes them practical for using. Opinions consist of three main classes: positive, neutral and negative. Opinions are often graded as different polarities corpus are very helpful for references, and feedbacks for governments, organizations in the adjustment of services to match customer's expectation (K. Dave, 2003).

Opinion classification was widely researched and developed an actual application in Chinese, France, and Japanese .etc. The applying fields are closely related to our research field such as restaurant evaluation. However, online hotel's service review has less attractive for conducting research in Viet Nam due to lack of training corpus. Recently, online hotel booking service and discussion are expanding rapidly in Viet Nam and availability of corpus, it is sound practicality for attempting a research.

Vietnamese language structure has a complicated phonetic structure that it contains 4 different kinds of tone marks such as rising tone " ' ", falling tone """, and the sentence structure is also different from other languages. That makes it is difficult to apply common studies on other languages for Vietnamese text classification. Unlike Western languages, in which blank spaces denote word delimiters, in Vietnamese, blank spaces play the roles of not only word delimiters

but also syllable delimiters (Diep, 2005; SCSSV, 1983) that cause difficulties in defining words. For example, the word “phòng không” can be tokenized as “phòng_không” literally translated as “air defense” or it can be tokenized as “phòng không” literally translated as “empty room”. In addition, part of speech tagging (POS) modern Vietnamese writing system is developed based on Latin system in which word is a combination of character and a representation of pronunciation, resulting in many homonyms, one word can act as noun, verb, or adjective. For instance, in the phrase “ông già đi”, “già” can be acknowledged as noun “old men walk, or father walks”, or as a verb “getting old”. Also, the word “đi” can reconsider as adverb “quickly” or verb “die”. Difficulties in Vietnamese occur in not only determining words as mentioned above but also bracketing phrases. One of the reasons is that there are many expressions having the same POS sequence but different phrase types in Vietnamese. Other difficulties are caused by the fact that word order in Vietnamese is very flexible.

1.3 Problem statement

From mentioned background information, there are many challenges we have to face when constructing a system for Vietnamese sentiment analysis. Especially for hotel reviews domain, we have a lack of training corpus, text preprocessing tool, and prior experiment results. Those problems have discourages researcher for attempting a research. As far as we know, there have is a previous study for sentiment classification from (Duyen N. T, 2014) based on available machine learning techniques. However, it is lexicon based study and the accuracy of the experiment is still below similar research for another language. It has not mentioned about how connectives such as *but*, *however*, *despite*, *although*, etc. are involved in contrasting discourse relations. In the realm of Rhetorical Structure, Theory found that the CONCESSION rhetorical relation was signaled by a connective 90% of the time in the newspaper article domain (M. Taboada. 2006). Therefore, the order of words and their sematic relations are essential and needed features for Vietnamese sentiment classification.

Based on the intensive survey, nobody has attempted to conduct Vietnamese sentiment analysis for hotel review based sentence dependency parsing and sub tree features. Most of the studies still have mentioned sentence syntactic structures that are vital need for the polarity of a whole sentence (Nakagawa, 2010). The major problem is that only a whole sentence is marked with its polarity by sentence composition and machine learning techniques. In turn, each individual element of the sentence is not labeled. Therefore, sentiment analysis based on dependency parsing with extracted sub trees is necessary to classify sentence polarity from its sub opinions.

1.4 Study objectives

The purpose of the thesis will address the following mentioned objectives:

- i. To Study the general techniques sentiment polarity classification and find a more efficient solution to improve the accuracy of Vietnamese sentiment classification.
- ii. To make a comparative study of Vietnamese sentiment classification based on its features and sentiment dependency parser.
- iii. To program by Java a completed Vietnamese sentiment classification system based on term features selection techniques and sentiment dependency parser.

1.5 Scopes of the study

The limitations and coverage of this thesis are presented below:

- The system can only classify sentence polarity, it could not identify a whole document or paragraph polarity.
- The thesis does not go in depth of building a Vietnamese parser, but focusing on the application of Vietnamese parser for sentiment classification.
- The developed program is applicable for Java programming language.

1.6 Thesis outline

Chapter 2: This chapter takes a survey on the previous experiment of sentiment analysis based on lexicon base with several common used machine learning techniques such as Decision Tree, Naïve Bayes (NB), Supported Vector Machine (SVM). Also, we take further look as feature selection techniques such X^2 (CHI), Mutual Information (MI), and Information Gain (IG). Finally, we conduct a survey on Dependency Parser such as non-projective parser and projective parser.

Chapter 3: We describes in detail about methodologies used for sentiment classification. The sections in this chapter describe feature selection method for Vietnamese langue, the software we used, and the data structure. Moreover, this chapter goes in depth of how we applied dependency tree for sentence classification.

Chapter 4: This chapter presents the result of proposed techniques and conducting a comparative evaluation. Other results such as the statistical are also provided in tabular form.

Chapter 5: This chapter provide the summary of the work done, statement of the conclusion, final results, and suggestion for possible extensions for a future revisit.

Chapter 2

Literature Review

This chapter presents the technical concept of term feature selection techniques and dependency parsing tree for sentiment analysis. The recent researches finding done for other languages with similar techniques. Also, this chapter provides scientific achievements of sentiment analysis with different techniques. In turn, this chapter presents the significance of our research.

2.1 Sentiment classification by features selection

Sentiment analysis is the procedure of investigating people's opinions, attitudes, and feeling toward certain things existed in the universe. In the past few years, this field has attracted a great attention from many endeavors due to a wide range of applications, and academic challenges (Jeevanandam, 2016). The sentiment reflecting opinions toward a particular product is irregularly expressed as positive or negative respectively. The opinions are often mixed with various features, some positive and some negative. The feature is made more sense than the overall opinion (Subhabrata Mukherjee, 2012).

Morgane Marchand (2013) presents the method of implementing semantic features and multiple polarity words for conducting sentiment analysis in twitter. They conducted their research on the corpus that contains marked words and phrases, the system has automatically evaluated whether a mentioned feature is positive, negative or neutral in that context. They developed a rebased research in combination with sentiment lexicons, and machine learning techniques. Initially, they refined the tweets based on words frequency from a sentiment corpus and then apply different supervised learning methods on the grounds of this initial classification. After that, they used different symbols (+,-,*) to denote a positive, negative and neutral tweet segment, respectively. Also, we use the \rightarrow b notation when referring to a polarity shift from a to b.

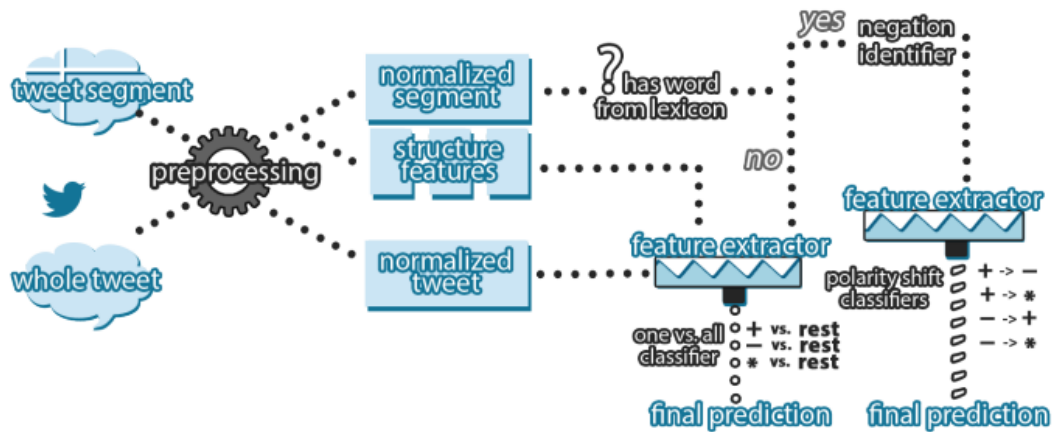


Figure 2.1: Contextual polarity disambiguation task system description (Morgane Marchand, 2013)

After segmentation of text by different mentioned polarities, the machine learning techniques were employed for automatically classification. The machine learning techniques used are Sequential Minimal Optimization (SMO), Random Forests and a Naive Bayes.

Table 2.1 presents a detail of their method performance.

Class	P	R	F-score
Twitter_positive	0.8623	0.9140	0.8874
Twitter_negative	0.8453	0.8086	0.8265
Twitter_neutral	0.4127	0.1625	0.2332
SMS_positive	0.7107	0.8945	0.7921
SMS_negative	0.8687	0.7609	0.8112
SMS_neutral	0.3684	0.0440	0.0787

Table 2.1: Performance of Morgane techniques

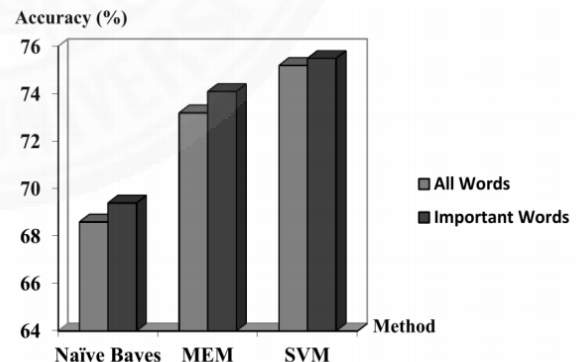
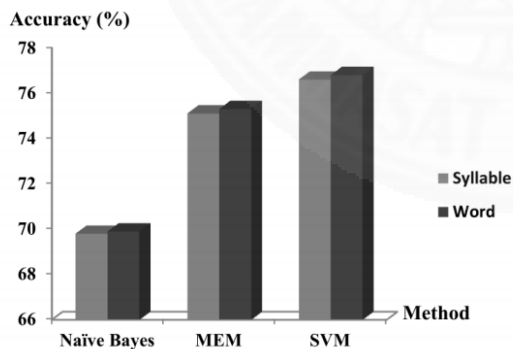
In addition, Daniel (2015) proposed a sentiment polarity classification using structural features. They investigated the role of contrasting discourse relations signaled by cue phrases, together with phrase positional information, in predicting sentiment at the phrase level. They implemented a research on two domains. The first domain is of nutritional supplement reviews, and the second domain is hotel reviews. They collected feature of discourse relations that describe how different segments discourse, or non-overlapping spans of text, interact. Studies have also examined how connectives such as but, however, despite, although, etc. are involved in contrasting discourse relations. When the corpus was ready, the author selected Java-based OpenNLP (M. Porter, 1980) toolkit for its maximum entropy classifier. The MaxEnt models were trained on approximately half of the reviews and tested on the other half; the training and testing sets were then flipped, and the results aggregated.

The system evaluated systems with the following features to train the models:

- **Baseline :**
 - Bag-of-words with negation handling, converting all words to lower case.
 - Raw-score of the segment.
 - The overall sentiment of the review, provided by the reviewer.
- **conj:** the discourse relation conjunction words beginning each segment, and the outcome of the prior segment
- **idx:** the position of the segment within the review, if it falls within the first 3 segments (for the supplements domain), or the first 6 segments (for the hotel's domain)

In Vietnamese language, Duyen .N.T (2014) investigated the task of sentiment classification by constructing machine learning model and selected language feature. They first constructed an annotation corpus by labeling sentiment components of hotel reviews followed by human judges and Kappa verification formula. Secondly, they conducted a study to observe how different feature could relate to the overall review of classification performance. Initially, they constructed a corpus that contains 3304 sentences, including 1980 positive sentences, 777 negative sentences, and 547 neutral sentences. For building learning models, they selected 3 methods including Naive Bayes, Maximum Entropy Models (MEMs), and Support Vector Machines (SVMs). For applying feature selection, the author developed a set of features:

- **Words:** Taking all words in a sentence for conducting research.
- **Important words:** Consider only words that are the main element in a sentence such as important verb, nouns, and headwords.
- **N-grams of words:** Try with different word grams to monitor the efficiency.
- **Syllables:** Consider all syllables words in a sentence.
- **Important syllables:** Care only syllables that have important meaning on sentence.
- **N-grams of syllables:** Try with different syllable grams to find the best fit.



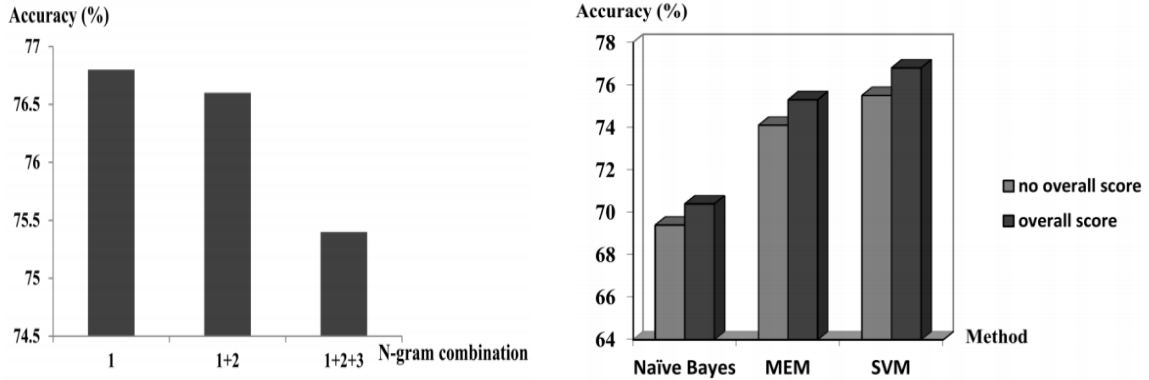


Figure 2.2: Comparison between proposed set of features (Duyen N.T., 2014)

From Dominique Ziegelmayr (2012) approach, statistical data compression as a non-standard method is deployed for sentiment polarity classification. The cross entropy $H(p, q)$ determines the normal data size per symbol associated with the certain event. The events are decided by possibilities based on programming scheme of given probability distribution q , rather than the true distribution p . Cross entropy for two probability distributions p and q over the same probability space is defined as:

$$H(p, q) := E_p[-\log q] = H(p) + D_{KL}(p||q)$$

where $H(p)$ is the entropy of p , and $DKL(pq)$ is the Kullback-Leibler divergence of q from p .

Their study was done on various measurement for cross entropy. These consisted of PPM, C_0 -measure, $C_{2.5}$ -measure, and $F_{2.5}$ -measure using n -gram frequency statistics. They tested the performance of those measurement techniques on two corpus including IMDB₁ movie reviews and Amazon₂ corpus. The table 2.2 below represents the performance of their technique based average accuracies of a ten-fold cross validation test.

No	Method	Accuracy	No	Method	Accuracy
(1)	PPMd	82.35%	(1)	PPMd	86.15%
(2)	C_0 -measure	83.10%	(2)	C_0 -measure	85.15%
(3)	$C_{2.5}$ -measure	84.90%	(3)	$C_{2.5}$ -measure	87.95%
(4)	$F_{2.5}$ -measure	85.30%	(4)	$F_{2.5}$-measure	90.55%
(5)	SVM (pres. unigram)	86.35%	(5)	SVM (pres. unigram)	86.35%

IMDb Corpus

Amazon Corpus

Table 2.2: The result of experiment

1 rec.arts.movies.reviews

2 amazon.com

Kieu and Pham (2010) proposed a rule-based method for constructing automatic evaluation of users' opinion at sentence level for the Vietnamese language. They built a system based on three components including preprocessing, dictionaries, and rules to classify opinion on a computer product. In preprocessing, they performed Vietnamese word segmentations and part of speech tagging. Dictionaries contain positive words and negative words. Lastly, the third component contains set of rules for word identification, sentence classification, and features evaluation. They applied Gate's Jape grammar to specify the rules. Rules are divided into 4 type:

- Pre-build dictionary for looking up and word correction.
- Word sense recognition.
- Sentential sentiment classification
- System for evaluating sentiment features

The sentiment classification is constructed by following procedures:

- Breaking sentence into clauses and sub-sentence
- Classify sentence into predefined categories such as positive, negative, mixed sense, and comparison sense.

They evaluated the performance of the system on each feature for all products. In this experiment, there were five features used for evaluation including operation, price, monitor, configuration, and outlook.

	#Annotation	#Annotation	#True annotation	Precision	Recall	F-measure
Pos Sen	231	218	154	70.64 %	66.67 %	68.60 %
Neg Sen	97	96	67	69.79 %	69.07 %	69.43 %
Mix Sen	9	26	7	26.92 %	77.78 %	40.00 %
All	340	343	231	67.35 %	67.94 %	67.64 %

Table 2.3: Performance result for Kieu research.

Another approach for Vietnamese sentiment analysis is mining of comparative sentences Ngo X. B. (2015) has conducted a research to take this rich feature for building a sentiment classification system. The research consists of building comparative sentences identification, and recognition of relations.

- **Comparative sentence identification:** this module receives a review sentence and identifies whether it is a comparative sentence or not. In the case that the input sentence is a comparative sentence, the module also classifies it as either equal, non-equal, or superlative comparison.
- **Comparative sentence identification:** this module receives a review sentence and identifies whether it is a comparative sentence or not. In the case that the input sentence

is a comparative sentence, the module also classifies it as either equal, non-equal, or superlative comparison.

- **Relation recognition:** this module receives an identified comparative sentence and recognizes entities, features, and comparing words in the sentence.

Model	Entity			Feature		
	Pre(%)	Re(%)	F ₁ (%)	Pre(%)	Re(%)	F ₁ (%)
Equative	95.78	82.35	88.56	83.33	63.39	72.00
Non-equative	95.10	91.35	93.19	83.80	65.50	73.53
Superlative	95.50	92.79	94.12	88.49	73.00	80.00

Table 2.4 Recognition results on three types of sentences

Table 2.4 compares experimental results between three sentence types, equative comparison, non-equative comparison, and superlative comparison

2.2 Sentiment classification by dependency parsing tree

A major problem associated with sentiment classification based on machine learning and its composition is that corpus is only labeled as sentence level, and a feature that is below sentence level is not labeled. However, those sub-feature are very important for supervised machine learning since the interaction of sentence composition can form sentence-level meaning. Dependency tree based method is a useful way to classify the sentiment polarity since it can utilize the sub-features.

Peifeng Li (2011) proposed a syntactic structure based that mined syntactic feature contained inside dependency tree. The flat features is then combined with dependency tree to form a novel feature representation for sentence-level sentiment classification. They used Stanford Parser3 which is indirect dependency tree. Each dependency relation is represented as a relation (*word1-location, word2-location*) and Stanford Parser currently supports 55 kinds of grammatical relations.

³ <http://nlp.stanford.edu/software/lex-parser.shtml>

Baseline and our approach	Features	Classifier
N-grams	Flat feature: unigrams (bag-of word)	Linear kernel-based SVM classifier
FT	Structured feature: Full syntactic parsing tree	tree kernel-based SVM classifier
FDT	Structured feature: Full dependency tree	tree kernel-based SVM classifier
FDT+POS	Structured feature: Full dependency tree with POS tags	tree kernel-based SVM classifier
PDT+POS	Structured feature: Pruned dependency tree with POS tags	tree kernel-based SVM classifier
PDT+ N-gram	Flat feature: unigrams+ Structured feature: Pruned dependency tree	Convolution tree kernel-based SVM classifier

Table 2.5: Classifier and Features used in experiment

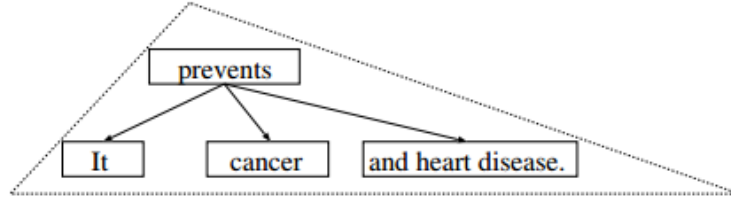
Approach	Positive			Negative		
	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
N-gram	0.6798	0.6889	0.6843	0.6796	0.6704	0.6750
FT	0.5832	0.6092	0.5959	0.5886	0.5453	0.5661
FDT	0.5952	0.6667	0.6289	0.6212	0.5467	0.5816
FDT + POS	0.5818	0.6134	0.5972	0.5792	0.5307	0.5539
PDT	0.6274	0.6756	0.6506	0.6684	0.6068	0.6361
PDT + N-grams	0.7452	0.7382	0.7417	0.7232	0.7376	0.7303

Table 2.6: The experimental result

The above result table indicates that mining syntactic feature in dependency tree has great advantages for sentence-level sentiment classification and F1 of FDT improves about 2.5% than that of FT.

Tetsuji Nakagawa (2010) took the advantages of applying dependency tree structure for sentiment analysis by using the conditional random field (CRF) with hidden variables. In his research, the sub-tree features in dependency are treated as a hidden variable because they are unobservable during labeling process. Sentence polarity was determined by those hidden variables based on CRFs.

Whole Dependency Tree



Polarities of Dependency Subtrees

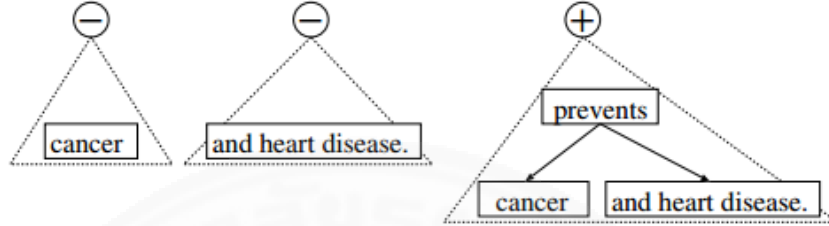


Figure 2.3: The sentence dependency tree with extracted sub-trees.

The joint probability is determined by given a subjective sentence w and its dependency tree h , using log-linear models:

$$P_{\Lambda}(s|w, h) = \frac{1}{Z_{\Lambda}(w, h)} \exp \left\{ \sum_{k=1}^K \lambda_k F_k(w, h, s) \right\}$$

$$Z_{\Lambda}(w, h) = \sum_s \exp \left\{ \sum_{k=1}^K \lambda_k F_k(w, h, s) \right\}$$

$$F_k(w, h, s) = \sum_{i=1}^n f_k(i, w, h, s)$$

where $\Lambda = \{\lambda_1, \dots, \lambda_K\}$ is the set of parameters of the model. $f_k(i, w, h, s)$ is the feature function of the i -th phrase.

$$p = \underset{p'}{\operatorname{argmax}} P_{\Lambda}(p'|w, h),$$

$$P_{\Lambda}(p|w, h) = \sum_{s: s_0=p} P_{\Lambda}(s|w, h).$$

The above equations show how to infer the sentiment polarity $p \in \{+1, -1\}$, given a subjective sentence w and its dependency tree h . They experiment was conducted in English and Japanese corpus. The below table shows the corpus metric of the experiment.

Language	Corpus	Number of Instances	(Positive / Negative)
Japanese	ACP	6,510	(2,738 / 3,772)
	KNB	2,288	(1,423 / 865)
	NTC-J	3,485	(1,083 / 2,402)
	50 Topics	5,366	(3,175 / 2,191)
English	CR	3,772	(2,406 / 1,366)
	MPQA	10,624	(3,316 / 7,308)
	MR	10,662	(5,331 / 5,331)
	NTC-E	3,812	(1,226 / 2,586)

Table 2.7: The corpus metric of the study

The result of a study with different methods and features used is shown below.

Method	Japanese				English			
	ACP	KNB	NTC-J	50 Topics	CR	MPQA	MR	NTC-E
Voting-w/o Rev.	0.686	0.764	0.665	0.727	0.714	0.804	0.629	0.730
Voting-w/ Rev.	0.732	0.792	0.714	0.765	0.742	0.817	0.631	0.740
Rule	0.734	0.792	0.742	0.764	0.743	0.818	0.629	0.750
BoF-no Dic.	0.798	0.758	0.754	0.761	0.793	0.818	0.757	0.768
BoF-w/o Rev.	0.812	0.823	0.794	0.805	0.802	0.840	0.761	0.793
BoF-w/ Rev.	0.822	0.830	0.804	0.819	0.814	0.841	0.764	0.797
Tree-CRF	0.846*	0.847*	0.826*	0.841*	0.814	0.861*	0.773*	0.804

(* indicates statistical significance at $p < 0.05$)

Table 2.8: Accuracy of Sentiment Classification

Shilpa Arora presented a novel representation of text based on patterns derived from linguistic syntactic annotation graph. They used a subgraph mining algorithm to automatically derive features as frequent subgraphs from the annotation graph.

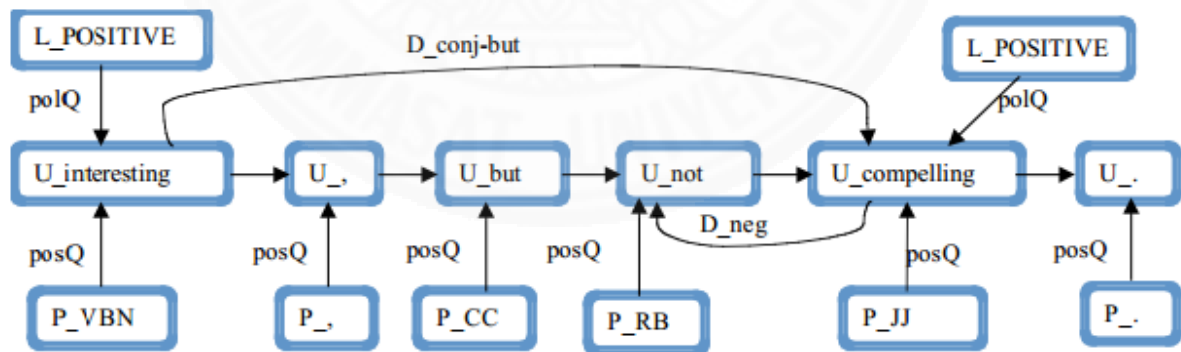


Figure 2.4: A syntactic annotation dependency graph constructed by Shilpa method.

Figure 2.4 presents three subgraph features extracted from the completed graph shown in figure 2.3. Figure 2.4 a shows the relation between words in the sentence, figure 2.4 b shows the polarity label of words, and figure 2.4 c presents the wildcards X on words that are polar of negating. Obtaining those graph feature, the author has performed a technique to find frequent subgraph patterns, from which they can construct features to use in the supervised learning algorithm.

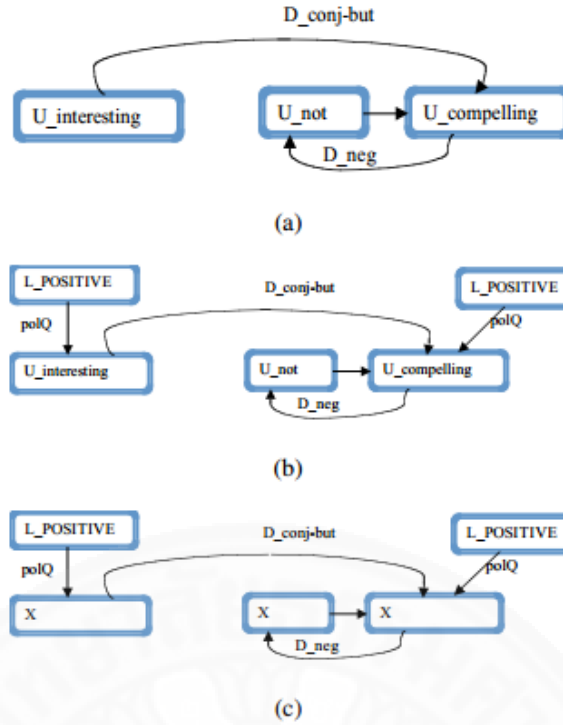


Figure 2.5: Sub graph features.

They have used 8000 sentences including 4000 positive reviews, and 4000 negative reviews from rotten Tomatoes corpus from (Pang and Lee, 2005). Unigrams (U), Part of Speech (P) and Dependency Relation Type (D) were used to label the node features. In addition, ParentOfGov and ParentOfDep were labeled for edge features. Supported Vector Machine (SVM) with linear kernel and the gSpan algorithm was set as a minimum for subgraph patterns to extract. As the result, they achieved 44, 161 feature with a factor of 5 increase in size. The classification result with different feature set is shown below.

Settings	#Features	Acc.	Δ
Uni	8424	75.66	-
Uni + Sub	44161	75.28	-0.38
Uni + Sub, χ^2 sig.	3407	74.68	-0.98
Uni + Sub, χ^2 size	8454	75.77	+0.11
Uni + Sub, (FS)	18234	75.47	-0.19
Uni + Sub, (Corr)	18980	75.24	-0.42
Uni + GP (U) †	8454	76.18	+0.52
Uni + GP (U+S) ‡	8454	76.48	+0.82
Uni + GP (U+S) †	8454	76.93	+1.27

Table 2.9: The classification result based on sub graph method

Ryan and Kevin (2006) figured the way to classify multilingual languages by applying two-stage discriminative parser. There were 13 diverse languages being tested in their study. In the first stage, they used a work done by McDonald and Pereira (2006) which was unlabeled dependency parsing models. This model contains morphological features and languages subset. The second stage inherited the result from the previous stage and annotate the edges by the

specific labels in the dependency graph with appropriate syntactic categories using a globally trained sequence classifier over components of the graph.

DATA SET	UA	LA
ARABIC	79.3	66.9
BULGARIAN	92.0	87.6
CHINESE	91.1	85.9
CZECH	87.3	80.2
DANISH	90.6	84.8
DUTCH	83.6	79.2
GERMAN	90.4	87.3
JAPANESE	92.8	90.7
PORTUGUESE	91.4	86.8
SLOVENE	83.2	73.4
SPANISH	86.1	82.3
SWEDISH	88.9	82.5
TURKISH	74.7	63.2
AVERAGE	87.0	80.8

Table 2.10: Results from an experiment conducted in 13 languages.

- **Unlabeled parsing:** Inheritance of work done by McDonald and Pereira (2006) using MIRA, an online large-margin learning algorithm to compute model parameters. It has the capability of processing a large amount of features over parsing decisions, as well as surface level features relative to these decisions.
- **Label classification:** Outputs from stage 1 is fed to stage 2, outputs parser y for sentence x and classify each edge $(i, j) \in y$ with a particular label $l(i, j)$.

The proposed system contain some elements such as the ability to produce non-projective edges, sequential averaged over Arabic, Bulgarian, Danish, Dutch, Japanese, Portuguese, Slovene, Spanish, Swedish and Turkish. N/P: Allow non-projective/Force projective, S/A: Sequential labeling/Atomic labeling, M/B: Include morphology features/No morphology features. The detail results are shown in table 1.6

SYSTEM	UA	LA
N+S+M	86.3	79.7
P+S+M	85.6	79.2
N+S+B	85.5	78.6
N+A+M	86.3	79.4
P+A+B	84.8	77.7

Table 2.11: Error analysis

Allowing non-projective parses helped with freer word order languages like Dutch (78.8%/74.7% to 83.6%/79.2%, unlabeled/labeled accuracy). Sequential did a slightly effect on overall label accuracy, but helpful in distingue the subject, object, and other dependents of the main verb.

2.3 Proposed experimental design

As an observation from above surveying, we could draw a problem that many types of research were just considering the features of a specific domain corpus and attempted to use those feature with available machine learning techniques. It was quite effervescent for product review evaluation since the features are limited. If the sentence has many dominant terms, then the system seem to classify the sentence based on high frequency recorded term. However, it is failed to classify the sentence based on the meaning of phrases or term. For instance, those system is unable to detect the terms that have a strong influence on the whole sentence. Also, the mentioned researchers have applied dependency tree structure to enriched sentiment features for better utilizing machine learning techniques. These studies have not touched the phrase relations such as negation, rewarded term, contradiction term etc. Also, this study will propose a comparative experiment on Vietnamese sentiment analysis based on term feature selection and dependency tree structure to solve those mentioned problems.



Chapter 3 Methodology

In this chapter, we present a system to classify Vietnamese online reviews based on term feature selection and dependency parsing. The system aims to classify up to sentence level based on three categories including “Positive”, “Negative” and “Neutral”. This is the 1st attempted research on the Vietnamese language based on the dependency tree.

3.1 System framework for Vietnamese sentiment analysis based on term feature selection

In an initial approach, we design a system follows figure 3.1. The system contains 5 modules, the first model handles a task of corpus collection from online hotel review source. The raw text collected from the 1st module will be going through a preprocessing module. This module basically recognizes a sentence, clean grammatically typos, phonetic checking, segmentation, part of speech tagging, and tokenization. In the next module, we develop a java program to extract a feature from the corpus, the detail will be described in the following section. The focus of this system will be the application of machine learning technique to classify Vietnamese sentences. Finally, we conclude the experiment by evaluation and recommendation module.

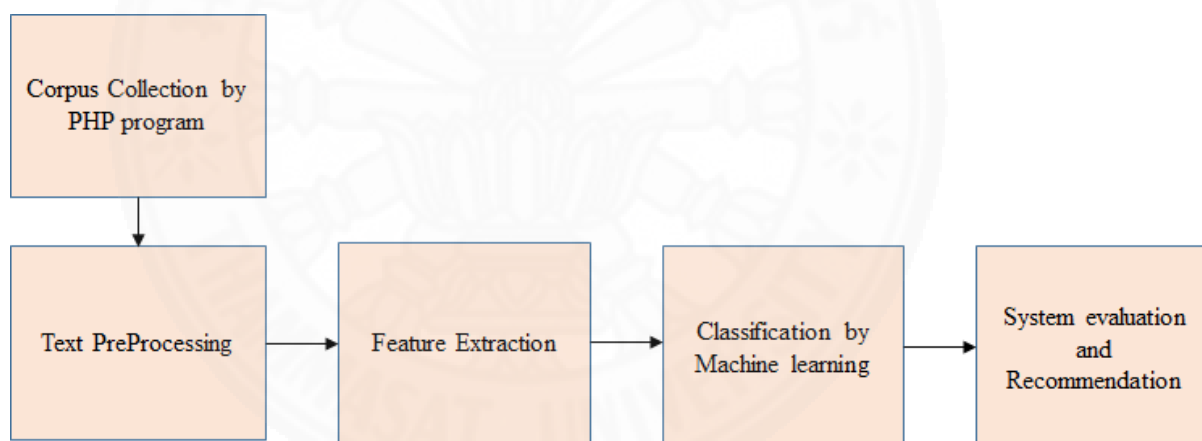


Figure 3.1: System overview.

3.2 Experimental data set

The research is primarily conducted in the Vietnamese language, therefore, the main driving source comes from Vietnamese websites. We built the corpus completely from scratch by developing a small PHP program to automatically collect Vietnamese comments on website name Agoda⁴. We collected the information from 10 most famous tourist attraction in Viet Name. Totally, there were approximately 300 hotels being visited. The general comment box consists of many features such as rating score on room quality, the value of money, convenience, location, sanitation, food, staff hospitality. The rating score is range from 1 to 10, however, based on our observation it's mostly felt to a range of 7.0 to 9.5.

⁴ <http://www.agoda.com/vi-vn/>

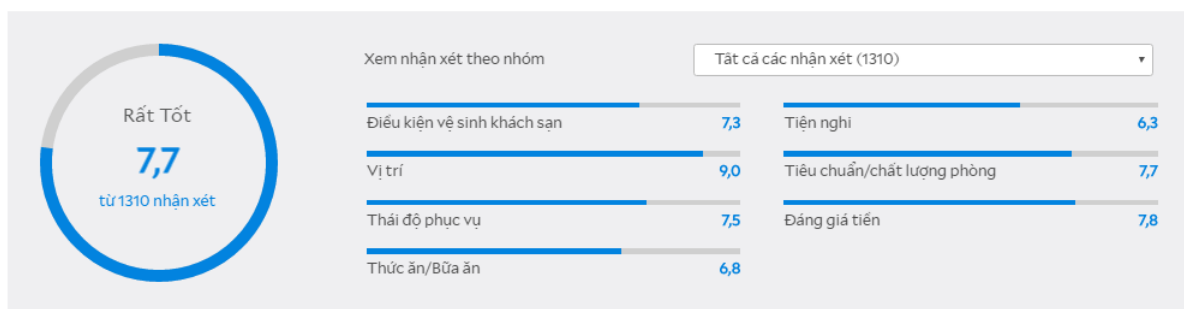


Figure 3.2: The general evaluation box (Agoda.com, 2016).

For each individual comment box, Agoda system also offers an individual rating score

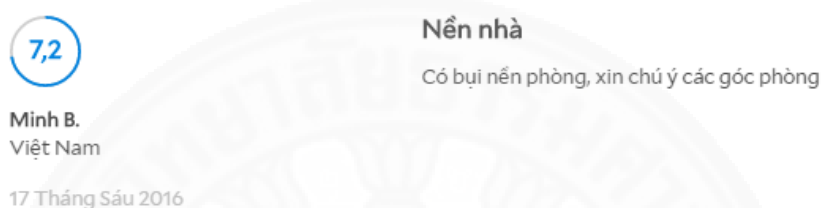


Figure 3.3: The individual comment box (Agoda.com, 2016).

When we executed our program, we only consider about rating score and the text. Only the sentence with a full stop at the end and written with correct tone mark will be collected. We also eliminated sentence with a single word or sentence with abnormal character. In the next step, we preprocessed sentences by Sentence Detection⁵ Part of Speech Tagging⁶ (POS), Word Segmentation, and Word Tokenization⁷.

⁵ <http://mim.hus.vnu.edu.vn/phuonglh/software/vnSentDetector>

⁶ <http://mim.hus.vnu.edu.vn/phuonglh/software/vnTagger>

⁷ <http://mim.hus.vnu.edu.vn/phuonglh/software/vnTokenizer>

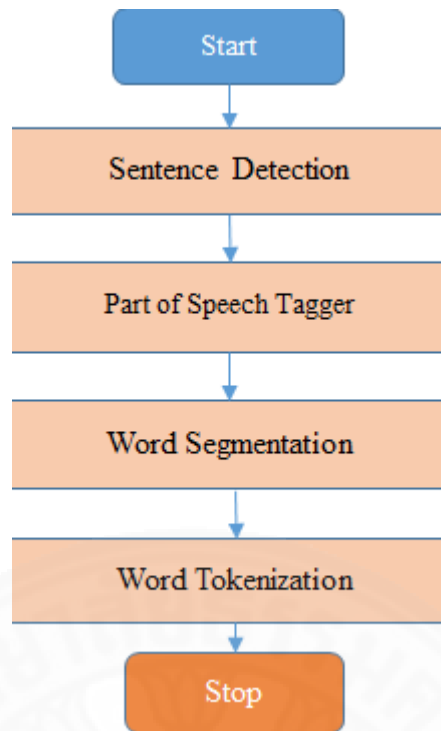


Figure 3.4: Main tasks in raw text pre-processing.

- **SentDetector:** The program was developed by Le H. P. (2008) based on maximum entropy approach. The program scans the sequences of texts that is separated by white space and contain the indicated symbol such as “.”, “?”, or “!”. These sequence of data was recorded as a *candidate* for training. The portion of the *candidate* preceding the potential character is called the *prefix*, and the portion following it is called the *suffix*. These are the features to detect whether a sequence of character is sentence or number such as *10.000 USD*. Based on those, feature the program will learn to distingue sentence boundary as valid or invalid.
- **vnTagger:** The models are train based on Maximum Entropy model and tested on part of speech tagged by Vietnamese Treebank. The Treebank built on a collection of 10, 165 sentences which are manually segmented, tagged, parsed. The main domains were social and political sections (Nguyen *et al.*, 2009). In tagging process, the maximum likelihood is assigned to the most frequent tag of the word sequence. The distribution p is chosen so that it has the highest entropy out of those distributions that satisfy a set of constraints.
- **vnTokenizer:** The technique is based on a hybrid approach to word segmentation of Vietnamese texts. The approach combines both finite state automata technique, regular expression parsing and the maximal matching method which is augmented by statistical methods to deal with ambiguities of segmentation (Phuong L. H, 2008). This tool was built with a Vietnamese lexicon that contains 40181 words which are commonly used language. The automata acceptor for the Vietnamese language consists of 42672 states, and 5112 are final states. In total, the system has 76249 transitions, the maximum outgoing transitions for 1 state are 85. Lastly, the maximum incoming transitions are

4615. The principle to decide segmentation rules are documents of ISO/TC37/SC 4 (2006).

After preprocessing the texts, the corpus was formatted as the table 3.1 below

<pre> </review> <-review Score="4,3" id="0"> <sentence Id="1" Class="NEGATIVE">Phòng/N nào/P cũng/R có/V muỗi/N và/CC kiến/N ./ </sentence> <sentence Id="2" Class="NEGATIVE">Rất/R nhiều/A muỗi/N ?? </sentence> <sentence Id="3" Class="NEGATIVE">Đồ_ăn/A thì/C dở/A ./ </sentence> </review> </pre>
<p>Translation</p> <pre> </review> <-review Score="4,3" id="0"> <sentence Id="1" Class="NEGATIVE">Every rom has mosquito and ant. </sentence> <sentence Id="2" Class="NEGATIVE">A lots of mosquito </sentence> <sentence Id="3" Class="NEGATIVE">The food tastes bad </sentence> </review> </pre>

Table 3.1: The format of Vietnamese corpus

The corpus has 1005 negative sentences, 501 negative sentences, and 676 neutral sentences. The corpus contain *review score*, *sentence Id*, and *class tags*. Sentence is confined in a tag `<sentence </sentence>`. Words are tagged with different annotation such as *P*, *V*, or *N* which indicated for *preposition*, *verb*, and *noun* respectively. In order to decide the sentences level tag, manually annotators were deployed. In the first step, each person was assigned sentence tags individually. In the second step, sentences with different tags are re examined to conclude a final tag. Cohen’s kappa coefficient is used to perform full disambiguation. We consider two parameters c_A and c_B agrees on category k : $P(c_A|k) \cdot P(c_B|k)$. A_e is agreement probability.

$$A_e = \sum_{k \in K} P(c_A|k) \cdot P(c_B|k)$$

The Cohen’s kappa coefficient was 0.89 which is a high reliability for perfect agreement.

3.3 Data format for experiment

We developed a program that is written by Java language, the program scanned through the whole corpus and recorded the frequency of every word. Only the words that appear more than 20 times in the corpus will be placed in a wordlist. Totally, 2969 keywords were successfully collected from out of 2182 reviews. The wordlist is sorted and formatted in a *hashtable* that contains word index and word frequency. The data structure is presented in a .arff format which can be read by Weka8 machine learning software. The arff format has two parts including a header section and a data section.

Positive Sentences	Negative Sentences	Neutral Sentences
1005	501	676

Table 3.2: The corpus metric

⁸ <http://www.cs.waikato.ac.nz/ml/weka/>

- **Header section:** This section has information about relation name, and a list of attributes (columns of data), and their data type. The detail of data structure is presented in table 3.1. @relation is written in a 1st line of arff file where <relation-name> is a string. The string must be quoted if the name includes spaces. @attribute declarations each attribute in the data set has its own @attribute statement which uniquely defines the name of that attribute and its data type. The @attribute has the form of:

@attribute <attribute-name> <datatype>

Where <attribute-name> has to start with alphabetic character. The <datatype> must fall into 4 support types in weka including numeric <nominal-specification>, string, date [<date-format>].

@relation data6.1.test	@data {1 1,3 1,8 1,21 1,27 1,28 1,47 1,51 1,60 1,118 1,1083 1,1202 1,2433 negative}
@attribute 0 numeric	{0 1,1 1,2433 positive}
@attribute 1 numeric	{0 1,1 2,3 1,5 2,8 2,47 1,51 1,105 1,160 1,162 1,1123 1,1353 1,2433 negative}
@attribute 2 numeric	{1 1,7 1,16 1,35 1,91 1,116 1,301 1,1933 1,2181 1,2433 positive}
@attribute 3 numeric	
@attribute 4 numeric	

Table 3.3: Header and data section of Arff format

- **Data section:** The declaration of data section begins with @data, attribute values are presented in series for each instance. They must follow after the header section. Explanation of sentence format with its attribute and frequency is shown in figure 3.3. The sentence data type is confined in the bracket ({}). Each word in the sentence is separated by a comma, and 1 pair of numeric data represents word index in the keyword list and word frequency. The last attribute represents the type of sentence which is pre-tagged manually by annotators.

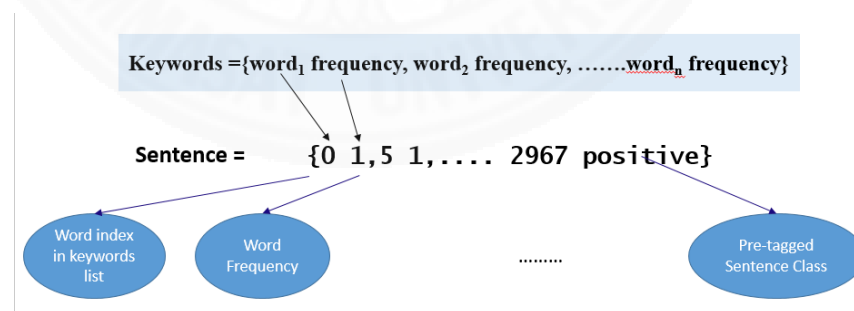


Figure 3.5: Structure of data section.

3.4 Text Classification Techniques

Text classification has a wide range of application in many contexts including document indexing based on a controlled vocabulary, to document filtering, automated metadata generation, word sense disambiguation, the population of hierarchical catalogs of Web

resources, and in general any application requiring document organization or selective and adaptive document dispatching (F. Sebastiani, 2002). There are several techniques have been used and achieved high performance in sentiment classification accuracy for other languages. For instance, K-Nearest Neighbor classifiers, Decision Tree, Bayesian classifiers, Support Vector Machines and Neural Networks are commonly used. Yang and Liu (1999) have conducted a survey on those mentioned methods, and in their comparative result SVM method was ranked as the best technique in term of its accuracy. The NB technique was the second runner, and Decision Tree was the least one. In order to perform a baseline comparison, we took Decision Tree, Naïve Bayes, and SVM to implement our study. In what follow, we will present the basic information on those techniques.

3.4.1 Support Vector Machine (SVM)

SVMs has gained attention in the machine learning and computer vision research communities (2008). It is only applicable for binary classification tasks, meaning that, using this method text classification have to be treated as a series of dichotomous classification problems. This is compatible with arff format that is perfectly working with a sequence of data. Classifier formally defined by a separating hyperplane, SVM classifies a vector d into either -1 or 1 using the following formula:

$$s = \sum_{i=1}^N \alpha_i y_i K(d, d_i) + b$$

3.4.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) algorithms sort the document's neighbors among the training document vectors based on their similarity which can be measured by for example the Euclidean distance or the cosine between the two document vectors. KNN is instance-based learning or lazy learning that does not have an off-line training phase. Therefore, it is considered as a simplest technique among other machine learning methods.

The kNN algorithm can be simply explained by given a test document, seeking for k nearest neighbors among the training documents, and uses the categories of the k neighbors to weight the category candidates [9]. The kNN can be written as:

$$y(\vec{x}, c_j) = \sum_{\vec{d}_i \in kNN} sim(\vec{x}, \vec{d}_i) y(\vec{d}_i, c_j) - b_j$$

where $y(\vec{d}_i, c_j) \in \{0,1\}$ is the classification for document \vec{d}_i with respect to category c_j ($y = 1$ for YES, and $y = 0$ for NO); $sim(\vec{x}, \vec{d}_i)$ is the similarity between the test document \vec{x} and the training document \vec{d}_i ; and b_j is the category specific threshold for the binary decisions.

3.4.3 Naive Bayes

The Naive Bayes (NB) algorithm was introduced by D. Lewis (1998). It is flexible that requires a number of parameters linear in the number of variables (features/predictors) in a learning problem. The conditional probability can be explained as the following formula:

$$p(C_k|\mathbf{x}) = \frac{p(C_k) p(\mathbf{x}|C_k)}{p(\mathbf{x})}.$$

3.4.4 Decision Tree (J48)

J48 is opening machine learning algorithm developing in the Weka⁹. Relying on a set of pre-labeled input data to build a decision tree based on C4.5. This algorithm was designed by Ross Quinlan.

3.5 Feature Selection Techniques

Feature selection can be subdivided into two areas which are supervised unsupervised. The supervised method will be involved with human supported in text data labeling. In the other hand, the unsupervised method will be conducted without the interference of human supports. In supervised feature selection, labeled training set of data will be modeled into the desired form. In the next step, the unlabeled test set will be analyzed to predict the outcomes. In contrast, unsupervised feature selection method does not require a pre-labeled dataset. But, heuristics learning algorithms are used for evaluation of the features.

3.5.1 Information gain

Information Gain evaluates the number of bits of information per category prediction by knowing the presence or absence of a word in at document (F.Sebastiani, 2002). Let c_1, L, c_k denote the set of possible categories. The information gain of a word w is defined to be:

$$\begin{aligned} IG(w) = & -\sum_{j=1}^K P(c_j) \log P(c_j) \\ & + P(w) \sum_{j=1}^K P(c_j|w) \log P(c_j|w) \\ & + P(\bar{w}) \sum_{j=1}^K P(c_j|\bar{w}) \log P(c_j|\bar{w}) \end{aligned}$$

3.5.2 χ^2 (CHI)

χ^2 (CHI): CHI is based on the statistical theory. It is useful in determining the statistical significance level of association rules. CHI is a normalized value and can be compared to the terms in the same category. CHI score between a term t and a class c is defined as:

$$\chi^2(t, c) = \frac{N \times [P(t, c) \times P(\bar{t}, \bar{c}) - P(t, \bar{c}) \times P(\bar{t}, c)]^2}{P(t) \times P(\bar{t}) \times P(c) \times P(\bar{c})}$$

⁹ <http://www.cs.waikato.ac.nz/ml/weka/>

3.6 Methods used

3.6.1 Baseline methods

In primary approach, we selected J48 learning algorithm classifier, Naïve Bayes, Support Vector Machines (SVM) to test with Vietnamese hotel review corpus. The techniques are available in Weka machine learning software, and 5 fold cross validation is applied for implementing the experiment. We split the corpus into 2 part, 80% of the corpus is used for model training, and 20% is used for conducting test case.

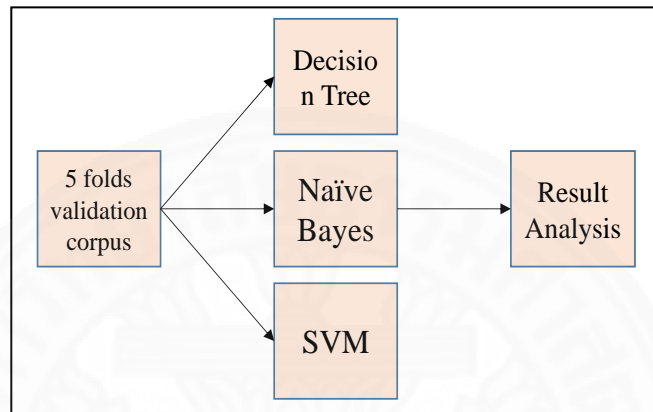


Figure 3.6: The framework for baseline experiment.

3.6.2 Machine learning and feature selection techniques

In this next setting, we applied the same classification techniques with the previous section, however, we added feature selection module before running classifiers. The feature selection module will preprocess data based on information gain (IG) evaluation and CHI (X_2) evaluation. Also, we split attributes into different of the amount of sets out of total 2434 attributes. The new system will be shown in figure 3.6

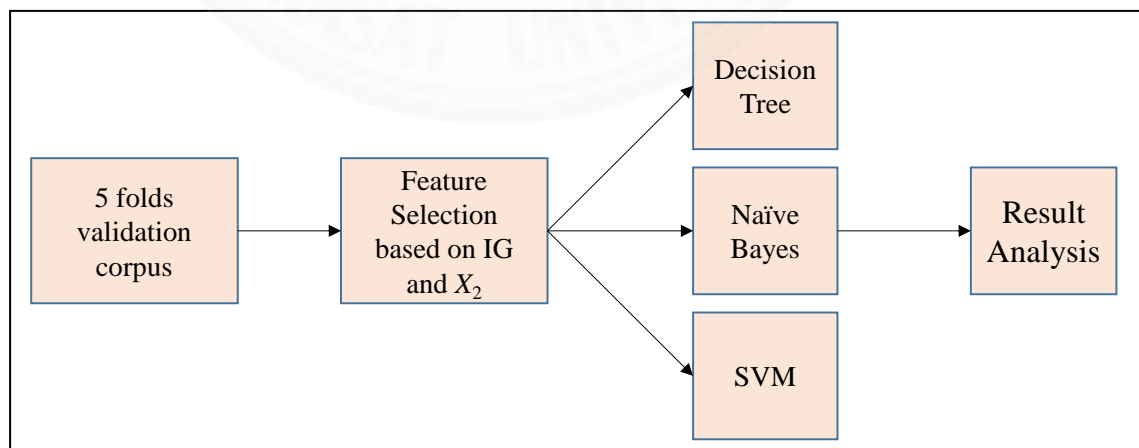


Figure 3.7: The new system with feature selection.

3.1 System evaluation method

Measurement of the experiment was based on recall, precision, and F-Score is used to evaluate the system’s performance. When we are comparing two annotations X and Y, these are:

$$\text{recall}(X, Y) = \frac{\text{number of identical nodes in } X \text{ and } Y}{\text{number of nodes in } X}$$

$$\text{precision}(X, Y) = \frac{\text{number of identical nodes in } X \text{ and } Y}{\text{number of nodes in } Y}$$

F-Score is concluded based on R and P:

$$F = \frac{2PR}{P + R}$$

3.7 Vietnamese sentiment classification based on dependency parsing

To explain our method, we consider a typical sentence “*The new window prevents mosquitos and flies but allows fresh air passing through*”. In this sentence “*mosquitos*” and “*flies*” are considered as negative polarities in hotel review domain. Our previous “Term features classification” which relied on counting word frequencies and applied machine learning techniques could wrongly classify whole sentence polarity as negative. Because, the polarities are reversed by modifying the word *prevents*, and the dependency subtree “*prevents mosquitos and files*” have positive polarity. In addition, the conjunction word “*but*” could link the 1st phrase “*prevents mosquitos and flies*” and 2nd “*allows fresh air*”. This conjunction word is considered as rewarded word that strengthens the positive polarity of the whole sentence because “*fresh*” has positive polarity.

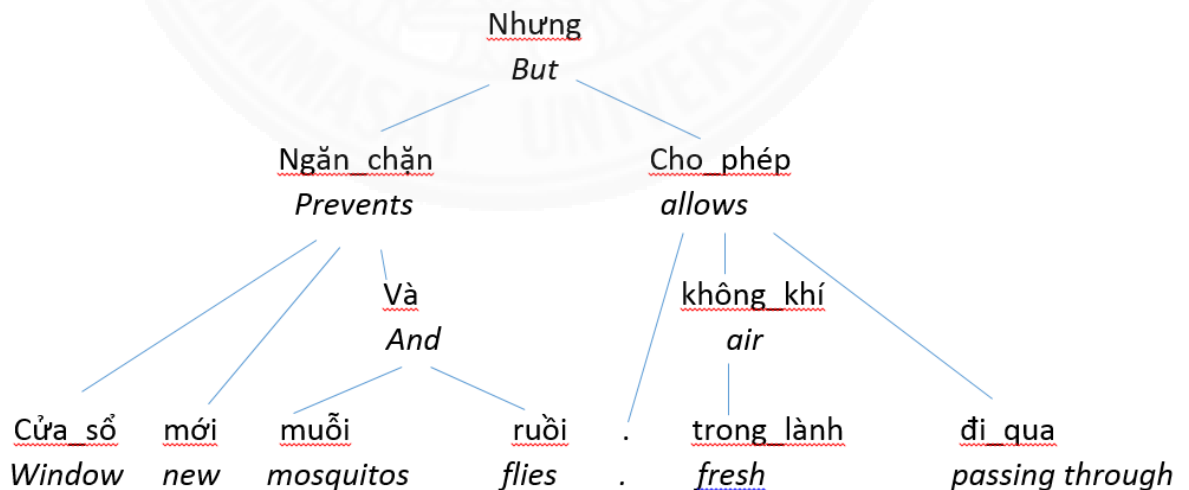


Figure 3.8: Original sentence dependency tree

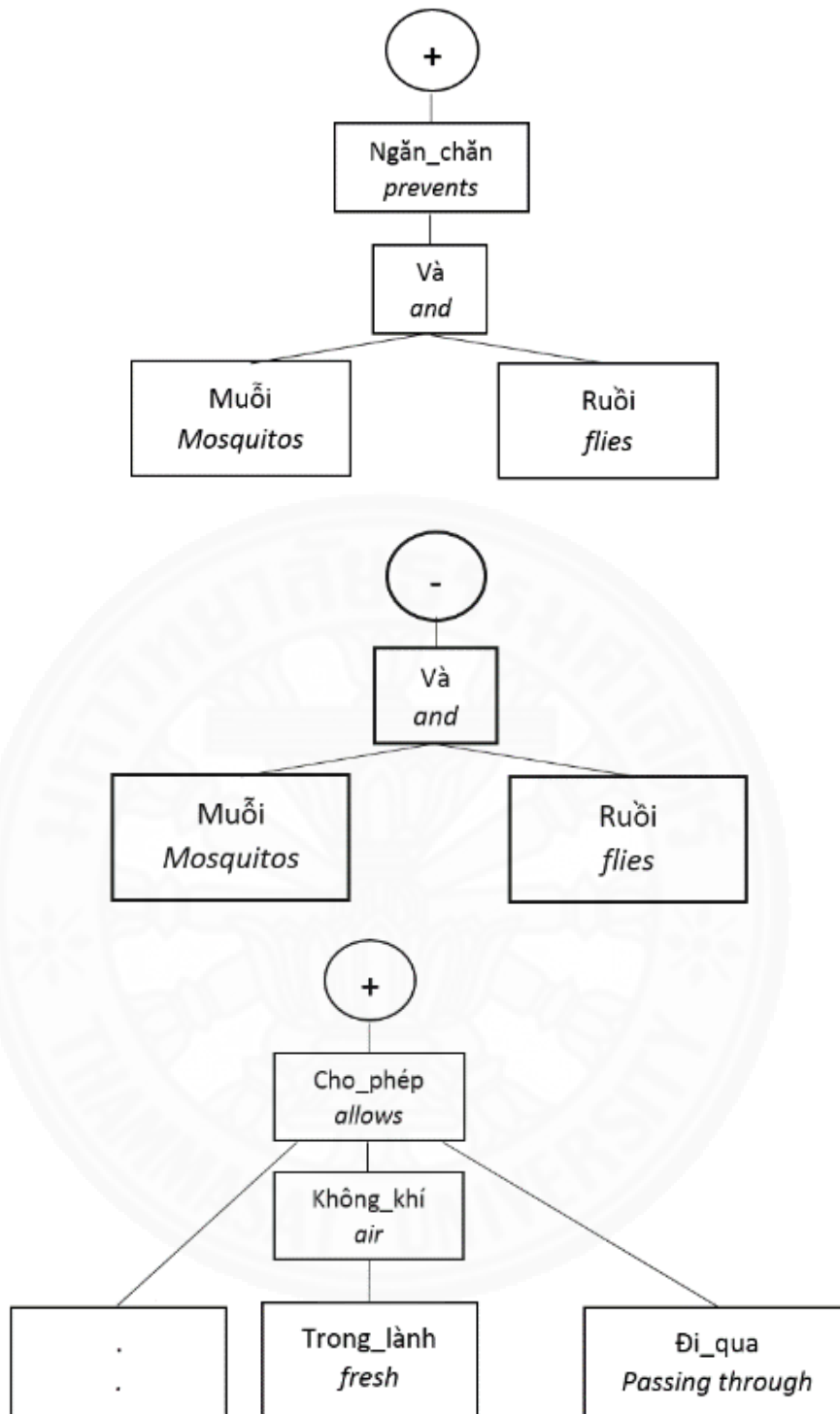


Figure 3.9: Polarities of Dependency Tree and Sub-Trees

In this manner, we can determine the sentence polarity based on dependency subtrees of a subjective sentence rather than considering each individual word. Because word phrases are more meaningful than words.

3.8 Syntax Theory

This section we analyze phrase and sentence structures based on Vietnamese language syntax. Syntax and Grammar

3.8.1 Syntax and Grammar

In order to construct a sentence, we must study about syntax. This tells us how to arrange elements in a sentence in a meaningful way. This is also a way a language can make a difference to another language. For instance, in English, the modifier (“red”) always precede a noun (“red label”) while in the Vietnamese language we write it after a noun (“nhãn can- label red “). Therefore, the syntax is an essential part in linguistic which must be prerequisite consideration of correlation between gesture and meaning. Vietnamese is an inflectionless language because its word form never changes, and there is no distinction in tenses.

Grammar is a relationship between syntax and morphology, and we often refer it as a complete set of rule that we can form a regular pattern in a specific language. Grammar can greatly help us to formalize words, sentences, phrases in a set of rules and patterns. There is two interrelated aspects in syntactic structure of sentences. The first one is phrase structure that concentrates about elements that form a sentence. The late one is dependency grammar that focuses on dependency relation. In this experiment, we will use this aspect to conduct our experiment.

3.8.2 Dependency structure grammar

In common sense, sentence can be ambiguous in a whole though individual word has a meaning. For example, the sentence “*Peter talk a nice bicycle*”, this sentence has no meaning a completed set, but individual word such as “*nice*” has a positive meaning. Therefore, sentence structure needs to consider about grammatical relations. Alternatively, it is termed as dependency structure since it covers dependency relation.

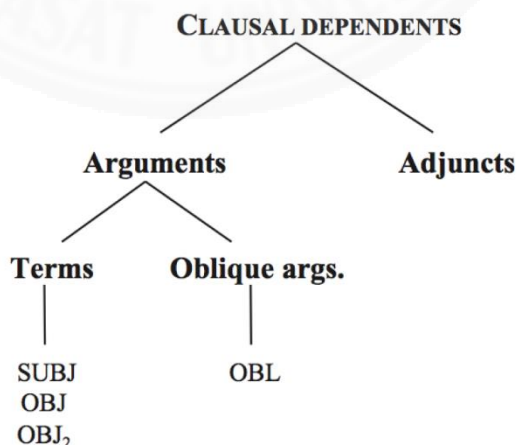


Figure 3.10: Sentence clausal element (Kroeger, 2005, p. 62).

3.9 New system approach

Figure 3.8 explains the new concept for Vietnamese sentiment classification. We kept the same corpus and text preprocessing module in the previous experiment. In next step, we parsed the sentences into a tree structure by applying projective dependency parsing. Subtrees extraction handles a job of breaking a whole sentence tree into a smaller size of the tree. The subtrees extracted from sentence tree are labeled with polarity dictionary, and from that, we decided sentence level polarity based on bottom-up manner. Finally, we do system evaluation based Precision, Recall, and F1 score.

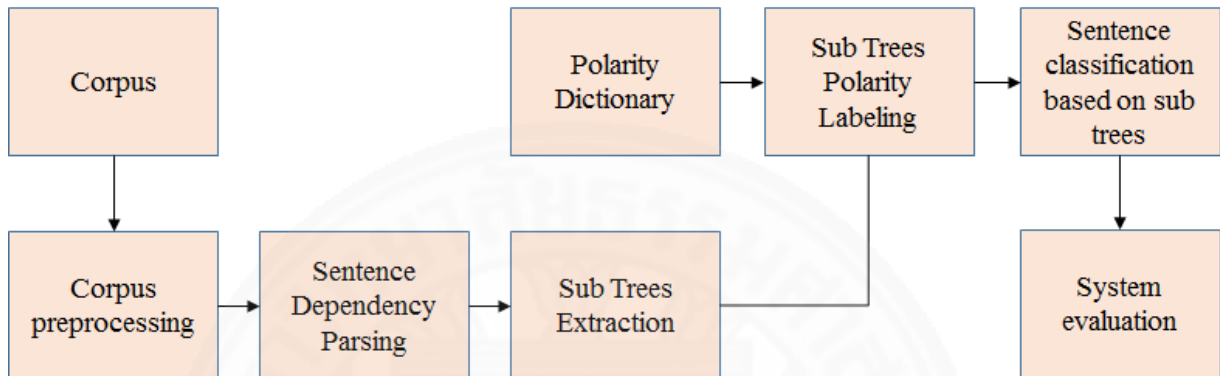


Figure 3.11: New system approach based on dependency parsing

3.10 Projective dependency parsing

Accordingly, parsing model that generates dependency graph representation of sentences is perfectly suitable for holding words and their relations. Words and their arguments can be modeled through directed edges, leaves, and nodes. Also, dependency graph contains rich features that can be further used for language processing. Those features were included in machine translation, sentence compression, and textual inference. Malt and MST are two available projective dependency parsers for conducting an experiment. From corpus collection observation, we detected that most of the reviews sentences are short and less than 30 words. In addition, the reviewers often break sentence without considering about the grammatical rule. Therefore, it is suitable to deploy MSTParser to construct dependency trees since it performed well on short sentences.

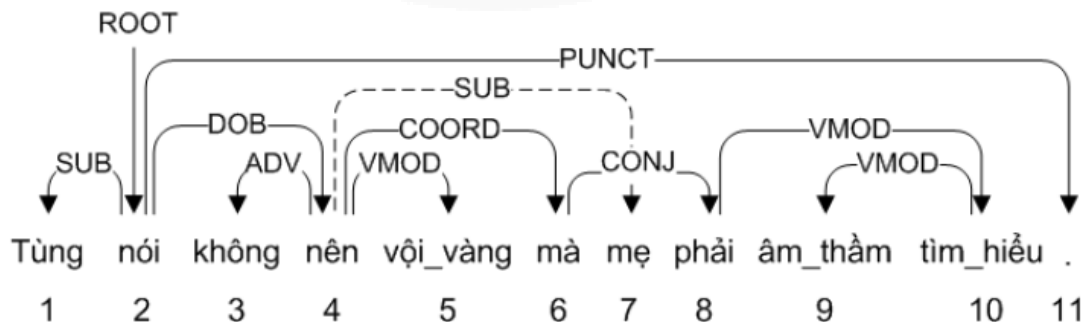


Figure 3.12: Vietnamese projective dependency tree transformed from Treebank.

Length	MST	Malt
<= 30 words	80.89	79.28
> 30 words	76.19	74.31
All	79.08	77.37

Table 3.4: Accuracy results of the parser.

We applied the vnDP model developed from Dai. Q. N. (2014) to handle this task. The model was constructed based on Vietnamese dependency Treebank VnDT which contains 10200 sentences. The dependency trees outputs are represented in form of CoNLL 10-column standard. The output contains node feature and edge feature, the root of the tree, and word tag.

The dependency tree is presented in the hierarchical dependency graph, it has a big advantage in the visualization of tree structure but it is difficult for computer reading data process. We decide to convert tree dependency graph into sentence bracketed form so that computer could easily read sub-tree inputs sequentially. We successfully developed an algorithm to bracket from dependency structure. The detail is shown in algorithm 1. From the CoNLL formate, we successfully bracket the phrases in the sentence so that we can extract them to be a subtree features.

Algorithm 1 Sentence Bracketing

Input: parent: list of parenthesis character parens=['(', ')']

tuple: a list of a tuple containing leaves and part of speech tagging. Order in the list presents the order of leaves in the tree structure.

Function sentence_bracket(self, parens)

 childstrs= "" # start with an empty string

for a child in self

if child is instance in Tree:

 childstrs.append(chil.sentence_bracket(parens))

else if child is instance in tuple:

 childstrs.append("".join(child))

else

 childstrs.append(child)

return parens[0], "".join(childstrs), parens[1]

1	cửa_sổ (window)	cửa_sổ (window)	N	N	-	3	sub	-	-
2	mới (new)	mới (new)	R	R	-	3	adv	-	-
3	ngăn_chặn (prevents)	ngăn_chặn (prevents)	V	V	-	0	ROOT	-	-
4	muỗi (mosquitoes)	muỗi (mosquitoes)	N	N	-	3	dob	-	-
5	và (and)	và (and)	C	C	-	4	coord	-	-
6	ruồi (flies)	ruồi (flies)	N	N	-	5	conj	-	-
7	nhưng (but)	nhưng (but)	C	C	-	3	coord	-	-
8	cho_phép (allows)	cho_phép (allows)	V	V	-	7	conj	-	-
9	không_khi (air)	không_khi (air)	N	N	-	8	dob	-	-
10	trong_lạnh (fresh)	trong_lạnh (fresh)	A	A	-	9	rmod	-	-
11	đi (pass)	đi (pass)	V	V	-	8	vmod	-	-
12	qua (through)	qua (through)	V	V	-	11	vmod	-	-
13	-	8	punct	-	-

Table 3.5: CoNLL format

(ROOT,ngăn_chặn cửa_sổ mới (dob,muỗi (coord,và ruồi)) (coord,nhưng (conj,cho_phép (dob,không_khi trong_lạnh) (vmod,đi qua) .)))

Figure 3.13: Bracketed sentence with semantic relation.

Sub-tree extraction module processes bracketed sentence and produce list of sub-tree as outputs. The sub-tree contain parent node, children node, and dependency relation tags. The statistical number of sub-tree and relation are illustrated in table 3.6.

Class	Number of sentences	Number of extracted subtree	Number of relations
Positive	2187	16304	33065
Negative	1826	13951	28751

Table 3.6: Corpus volume and extracted features

```

1. Và (ruỗi)
And (flies)
2. muỗi (và)
mosquitos (and)
3. không_khí (trong_lạnh)
air (fresh)
4. đi (qua)
pass (through)
5. cho_phép (không_khí đi .)
allows (air pass .)
6. nhưng (cho_phép)
but (allows)
7. ngăn_chặn (cửa_sổ mới muỗi nhưng)
prevents (window new mosquitos but)

```

Figure 3.14: Subtrees extracted from bracketed sentence

From this format, we can extract the phrases as mentioned in figure 1. The prior polarity of a phrase $q_i \in \{+1, 0, -1\}$ is the innate sentiment polarity of a word contained in the phrase, which can be obtained from sentiment polarity dictionaries. Since Vietnamese polarity dictionary was not available for hotel review domain, we decided to construct once by scanning the whole corpus and record the high frequency of occurrence words to initiate the polarity dictionary. In turn, the resulting dictionary contains 324 positive expressions and 332 negative expressions. Based on the edge feature in the dependency tree, we were able to construct *reversed* and *rewarded* expression dictionaries. As the result, there were 37 reversed expression and 13 rewarded expression successfully collected respectively.

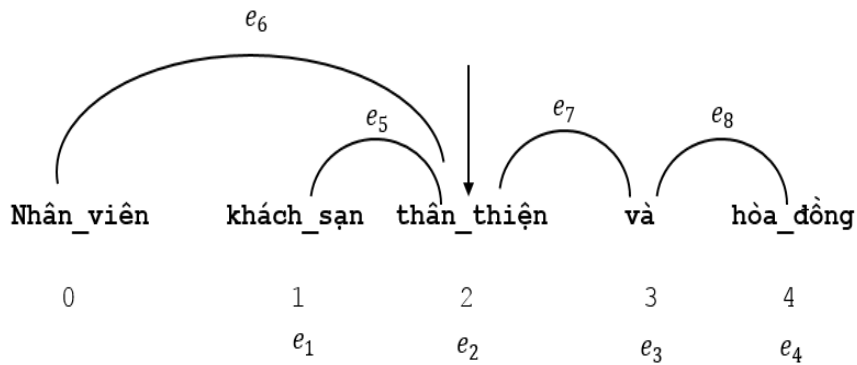
Sentence classification module categories sentence polarity based on its dependency sub-trees. Section 3.2 will describe in detail of classification methodology.

3.11 Classification with sub opinion relation

MacKay (2003, chapters 16 and 26) presented a theory of belief propagation, a generalization of the forward-backward algorithm that is deeply studied in the graphical model's literature (Yedidia et al., 2004). Belief propagation is well known as sum-product message passing, it calculates the marginal distribution for each unobserved node, conditional on any observed nodes. Belief propagation is commonly used in artificial intelligence and information theory and has demonstrated empirical success in numerous applications including low-density parity-check codes, turbo codes, free energy approximation, and satisfiability. A graph containing nodes corresponding to variables V and factors F , the edges connect variables and the factors. The joint mass function is:

$$p(x) = \prod_{a \in F} f_a(x_a)$$

where x_a is the vector of neighboring nodes to the factor node a . The function works by passing belief message in the edge of hidden nodes. Specifically, if node a is connected to node v in the dependency graph, a message denoted by $\mu_{(v \rightarrow a)}$ is passed from v to a and $\mu_{(a \rightarrow v)}$ is passed from a to v . The messages is computed differently based on whether a node is a variable node or factor node.



Translation: *The staffs are very friendly and hospitable*

Figure 3.15 Node features and edge features in dependency tree

Figure 3.11 represents dependency graph with variable nodes from 0 to 4, factor node is from e_1 to e_4 and edge features are from e_5 to e_8 . The technique start with passing a message from the leaves of a tree to their parent's node, when the parents belief is updated the message will be continuously passing up until it reaches to the root node. In the second process, another message is passing outward from the root of dependency tree to their leaves. The process keep running until every node belief is updated.

3.12 Classification with sub-relation

When a subtree contains the opinion relation word registered from dictionaries its polarity resulting from calculating from its leaves will be reversed. For instance, since "prevent" is reversed meaning it will change polarities of "mosquitos" and "flies" from negative to positive.

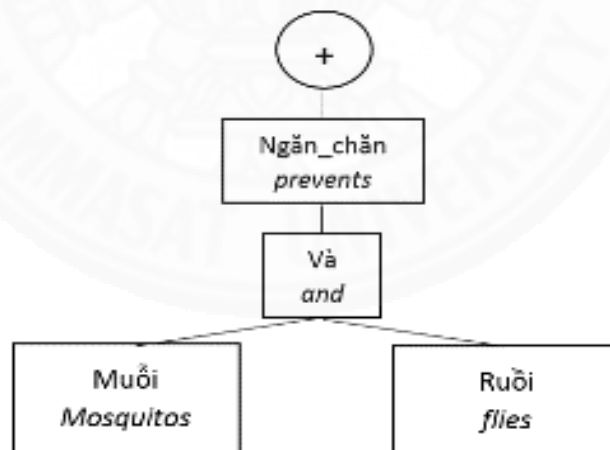


Figure 3.16: Calculating subtree polarity.

Therefore, the subtree polarity will be positive, and we also update the polarity of "prevent" to be positive which is useful to determine the subtree polarity in a higher level. From figure 1, we defined "but" as *contradiction meaning word* because it strengthens the polarity of its following phrase. Accordingly, if the sentence has a structure like:

(A but B)

When A is preceding phrase and B is proceeding phrase. We decide the sentence polarity as B polarity.

Sentence polarity is decided as the polarity of *B*. Algorithm 2 show a completed procedure for determining a sentence polarity that contains *reversed* and *rewarded* relation.

Algorithm 2 Sentence analysis with *reversed* and *rewarded* relations

Input: bracketed sentences

Function sentence_analysis(string FileName)

while has line and line is not empty

 find a pattern **m** which is a matched parenthesis pair

while **m** is found

 sub_tree= an element in group of pattern **m**

if a group of sub_tree does not contain sub-tree

 group_sub_tree.add(sub_tree)

for each sub_tree in group_sub_tree

if the sub_tree parent node is *reversed* relation

 sub_tree polarity is reversed polarity of its children polarity

 update parent_node polarity

else if the sub_tree parent node is *rewarded* relation

 sub_tree polarity is equal to polarity of its right children

 update parent_node polarity

else

 sub_tree polarity is decided by sum product propagation

 update parent_node polarity

return sentence_polarity

3.13 Classification with considering word granularity

It is a difficult task to determine an appropriate granularity of a word in a different concept. The Vietnamese language does have a concept of word tense for a verb like “run” in present tense, and it will be “ran” for past tense. Also, in English, most nouns need to be in a form of singular or plural (eg. pen vs pens) whereas Vietnamese nouns “do not in themselves contain any notion of number or amount” [34]. For instance, “bể bơi quá đông” (the swimming pool is very crowded) and “Khách sạn vào mùa đông khá vắng” (Hotel in winter is quite deserted). The word “đông” is used without changing its form. In the first sentence it has a form of the adjective, but in the second sentence, it has a form of the noun. Therefore, Vietnamese is isolating language that word formation is a combination of isolated syllables [35]. These syntactic aspects will be represented by using constituency of syntactic structures. This concept was implemented in the attempt of building Viet Treebank [30\6]. Viet Treebank consists of a corpus with word segmentation and POS annotation. vnDT model was constructed based on Viet Treebank that can provide a syntactic dependency of word sense by giving its most plausible syntactic analysis [30]. There is strong dependency of parent node word and its dependents. Thus, we can determine the sense of dependency based on syntactic structure. Word sense can be disambiguated at the granularity level of first sense. In order to utilize word sense with dependency tree, we use Vietnamese Wordnet [37]. Wordnet is an ontology that holds relationship among words and words senses, and words are organized in hierarchies of senses. In Vietnamese Wordnet, there are three main classes including of Synset, Word and WordSense.

Nouns, verbs, adjectives are related by the hypernym-hyponym relationship in which they are classified into a group of first sense. The complete set of first senses granularity is presented below.

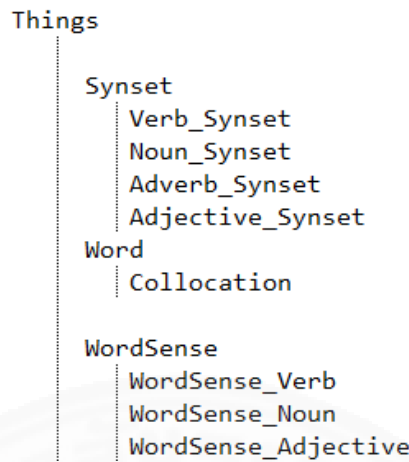


Figure 3.17: VietWordNet hierarchical structure

We constructed a word sense dictionary that has total of 778 words on different levels. Now the phrase polarity is not simply a set of $\{+1, 0, -1\}$ but rather fall in the middle of 0 and 1.

We will investigate a sentence:

Nhân viên đá vào hành lý
(Staff kicks the luggage)

This sentence is express as Figure 9:

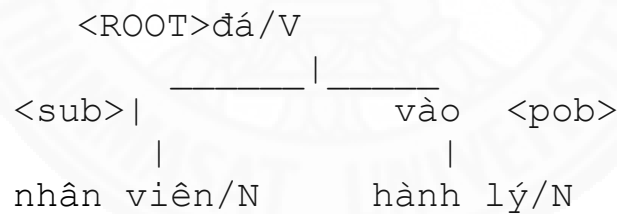


Figure 3.18.: An example of expressed sentence.

A word can hold multiple senses however in our research scope, we only address two major senses. In Table 3.7 we show a possible number of senses for above sentence.

đá (V) #1	kick, kicking, throwing, throw in, shot at, kicked
đá (V) #2	push, shove, push, nudge
đá (N) #1	rock, stones
đá (N) #2	ice, iceberg
hành_lý (N) #1	luggage, baggages
hành_lý (N) #2	personal things
Nhân_viên (N) #1	staffs, officer
Nhân_viên (N) #2	operators, a person works in an organization

Table 3.7 Sense for đá and hành lý

In intuitive step, we could assume that a word “đá(N)#1” (stone) is related to first sense “vật” (object). Also, a word “hành lý(N)#1” (luggage) is referred to the non-human thing, and its first sense is strongly related to “vật” (object). Thus, a combination (“đá(N)#1” (stone), “hành lý(N)#1” (luggage)) can be an appropriate sense combination. However, “Nhân_viên (N) #1” (staff) is referred to human, and “đá(N)#1” (stone) is strongly involved in (object). Therefore, a combination (“Nhân_viên (N) #1” (staff), “đá(N)#1” (stone)) has a weak sense agreement. We could form much possible sense combination following a set of senses in table 3. Finally, we could obtain (“Nhân_viên (N) #1” (staff), đá (V) #1 (kick), hành_lý (N) #1 (luggage)) is most suitable sense because đá (V) #1 (kick) has first sense in set of action that effect on object things “luggage”. In this manner, we can figure out that “đá” (kick) is most likely classified as a negative verb. After we disambiguate word sense, a technique in previous sections will be applied continuously to classify sentence polarity.

3.14 System evaluation method

Table 3.8 present the case that a sentence might be felt in when we conduct sentiment classification. This table is called confusion table consist of 2 rows and columns, and each cell indicates each type of prediction.

	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Table 3.8: Confusion matrix table

- **Precision:** This score is evaluated by measure the ratio of correctly predicted positive sentence over total of true positive and false positive. It is also called the Positive Predictive Value (PPV).

$$Precision = \frac{tp}{tp + fp}$$

Where *tp* is true positive, *fp* is false positive

- **Recall** can be measure by the following formula. It is a ratio of true positive over the total of true positive and false negative sentences.

$$Precision = \frac{tp}{tp + fn}$$

Where *fn* is false negative

- **F1 Score:** A single measure that trades off precision versus recall is the F-measure, which is the weighted harmonic mean of precision and recall

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad \text{where} \quad \beta^2 = \frac{1 - \alpha}{\alpha}$$

The default balanced F-measure equally weights precision and recall, which means making $\alpha = 1/2$ or $\beta = 1$. It is commonly written as F1, which is short for $F_{\beta=1}$.

$$F_{\beta=1} = \frac{2PR}{P + R}$$

Put another way, the F1 score conveys the balance between the precision and the recall.

- **System accuracy:** is measured by the following formula

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

Where *fp* is false positive

3.15 Summary

In this chapter, we described the methodologies used to conduct our research. In the 1st stage, we implement Vietnamese hotel review sentiment analysis by feature selection with machine learning techniques. In the 2nd stage, we changed the method by switching to dependency parsing. Vietnamese Malt parser developed based on Vietnamese TreeBank, word polarities, and extracted subtrees were brought together to determine sentence-level polarities. In the next chapter, we will present the comparative results and make an analysis.

Chapter 4

Result and Discussion

This chapter presents the result of proposed methods in chapter 3, the results are organized in order including feature selection method, dependency parsing method, and comparative summary of two methods.

4.1 Results and Analysis for Term Feature Selection

Table 4.1 shows the result of sentiment classification by Decision Tree, Naïve Bayes, and SVM. In overall, we obtained that Naïve Bayes delivered the highest performance in all classes, the highest result is 91.8 % in “POSITIVE” class based on Recall. The highest result of SVM method was 87.8% in “POSITIVE” class based on Recall. The highest result for Decision Tree method was 82.3% based on Recall.

Methods	Precision	Recall	F-Measure
Decision Tree			
POSITIVE	0,712	0,823	0,764
NEGATIVE	0,5	0,441	0,469
NEUTRAL	0,67	0,574	0,618
Weighted Average	0,65	0,658	0,651
Naïve Bayes			
POSITIVE	0,698	0,918	0,793
NEGATIVE	0,52	0,411	0,459
NEUTRAL	0,765	0,525	0,623
Weighted Average	0,678	0,68	0,664
SVM			
POSITIVE	0,725	0,878	0,794
NEGATIVE	0,628	0,481	0,545
NEUTRAL	0,67	0,577	0,62
Weighted Average	0,686	0,693	0,683

Table 4.1: The result of sentiment classification.

Figure 4.1 shows the average result of three selected method. The SVM got the highest place that its accuracy was 69.3%, Naïve Bayes success rate was 68%, and Decision Tree has lowest performance as its accuracy was 65.8%.

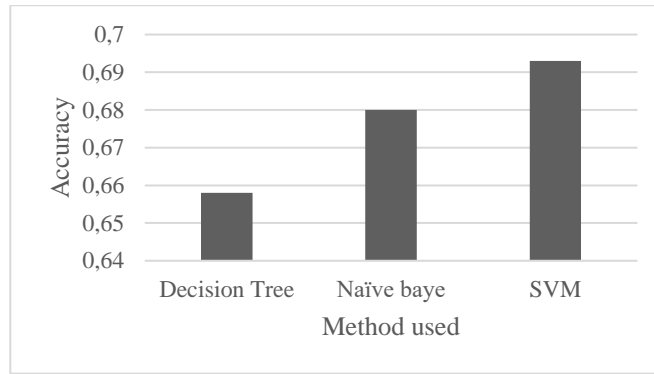


Figure 4.1: The accuracy of the used methods

The feature selection technique was handled by the Weka, the data mining software that allowed us to adjust the number of attributes in the preprocessing data. Information gain (IG) and χ^2 (CHI) were applying in preprocess phrase in Weka. The number of attributes was selected from 2969 keywords. We run the test case on a different number of attributes ranging from 240 to 1200.

Figure 4.2 shows the result of sentiment classification when we applied information gain. In overall, SVM delivered the best result in comparison with Naïve Bayes and Decision Tree. In Precision measurement, the highest accuracy of SVM was 71.4% while Naïve Bayes was 68.8%, and Decision Tree was 65.4% respectively. The same scenarios happened in Recall measurement when SVM got the highest performance with 71% of accuracy. Naïve Bayes and Decision Tree had 68.5% and 65.8% accordingly. Lastly, in F-Score measurement SVM had its highest accuracy of 69.3%, Naïve Bayes had the second place with an accuracy of 66.6%, and the lowest was Decision Tree with an accuracy of 65.1%.

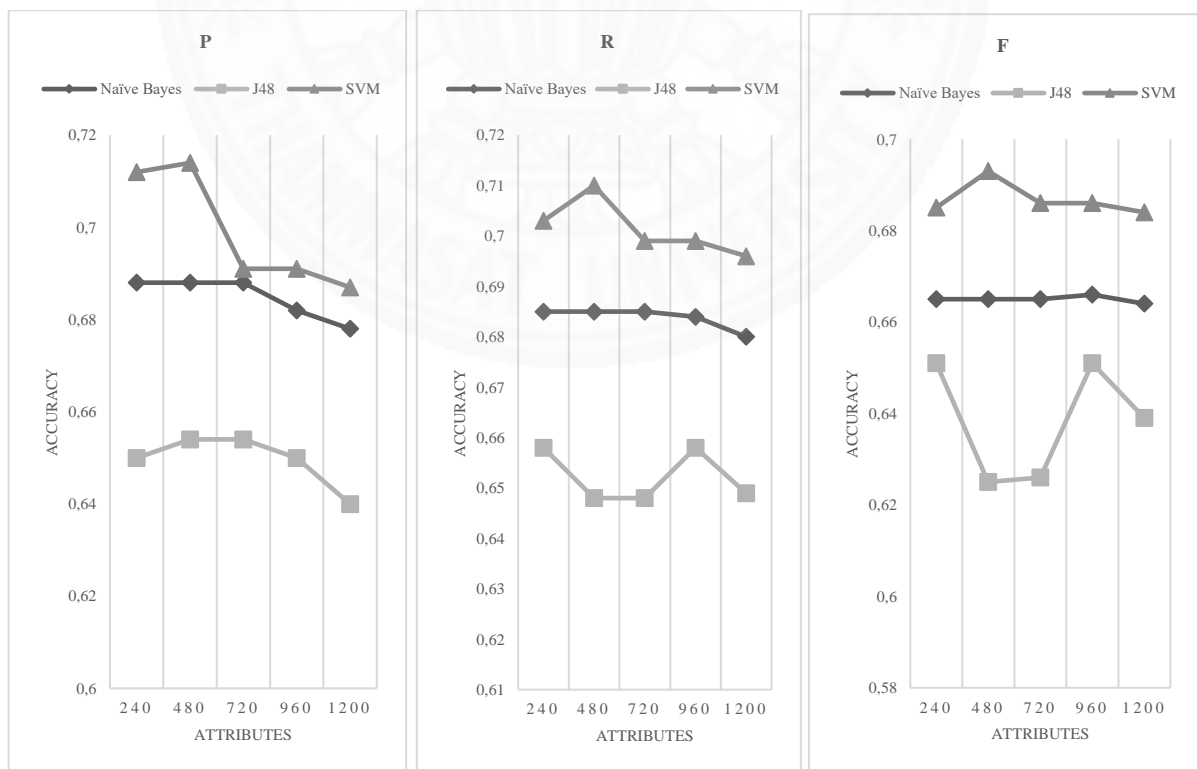


Figure 4.2: The result with Information Gain feature selection

Figure 4.3 present the result of sentiment classification with application of CHI square feature selection. As we observed, the overall performance was slightly improved when we applied Information Gain feature selection technique. Decision Tree delivered the best result of 78.4 % in F-Score measurement with number of attributes are 1200. SVM has second highest accuracy which were 71.4%, 71%, 69.3% in Precision, Recall, and F-Score measurement respectively. While, Naïve Bayes has 69.7%, 69.3%, and 67.3% separately. Finally, Decision Tree has lowest performance when the accuracies were 65.6%, 65% in Precision and Recall. Although, we witnessed that Decision Tree has highest performance but throughout whole process with different number of attributes Decision Tree has lowest performance. This result confirmed that our experiment achievement agreed with other studies from other languages.

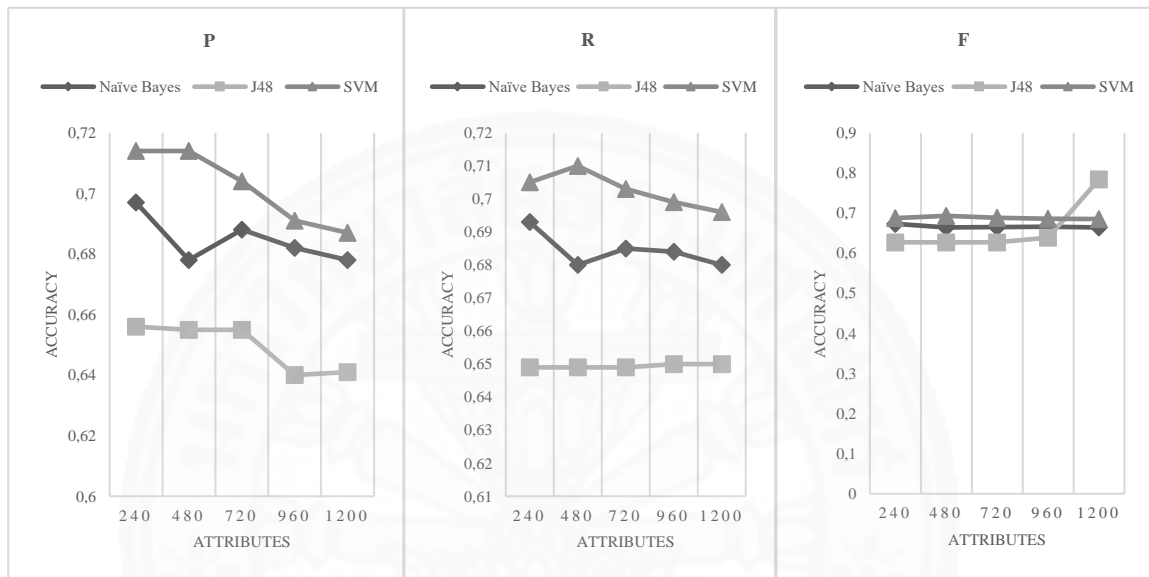


Figure 4.3: The result with CHI square feature selection.

During the experiment, the recorded data showed that the accuracy POSITIVE term usually had the highest accuracy. For example, it was 94.9% in IG feature selection, and 95.2% in CHI feature selection. This can be explained that the number of positive samples (1005) is higher than negative samples (501) in the corpus. Another reason is positive sentences are usually stated clearly, while negative sentences are often stated implicitly.

4.2 Results and Analysis for Sentence Dependency Parsing

We compared the result of a new experiment with the previous experiment which is carried based on feature selection and machine learning technique. The detail of results is shown in table 4.4 below.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Feature selection with Decision Tree	65.3	65	65.8	65.1
Feature selection with Naïve Bayes	67.4	67.8	68.0	66.4

Feature selection with SVM	68.73	68.6	69.3	68.3
Tree and sentence bracketed with Sum-production propagation	71.90	90.59	68.26	77.85
Tree and sentence bracketed with rewarded and reversed voting	86.04	87.97	86.63	87.30
Tree and sentence bracketed with Word Sense	89.60	90.26	90.63	90.45

Table 4.2: Comparative result among used methods

4.2.1 Tree and sentence bracketed with Sum-production propagation

This technique represents a result of using tree structure in which words with the same level in sentence tree structure were bracketed together. After that, we apply sum-product propagation technique to calculate an overall sentiment polarity score. The score archived by this technique is the main factor to decide polarity of tested sentence. From the table 4.2, we can see that the accuracy is just slightly improved from machine learning technique. The big difference was shown by precision measurement, however, this is a fault result due to the fault negative recognition. The fault negative causes by applying technique can not recognize negation relationship between words, and terms in a sentence.

4.2.2 Tree and sentence bracketed with rewarded and reversed voting

In this experiment, we substituted word and term relation together with sentiment tree structure for evaluation. We defined *rewarded* and *reversed* relation by specifying connection word in a sentence. We can look back in methodology chapter for more detail of the technique. The result in table 4.2 clearly shown the improvement of this technique. The achievement is much better than the combination of feature selection and machine learning techniques. From this table, the accuracy is 86.04% while SVM can only archive accuracy of 68.73%. This big difference is reasonable because common machine learning technique like SVM merely pays attention to rich extracted feature from corpus but skipping relation of words and terms. The best measurement result gained by precision measurement which was 87.97%.

4.2.3 Tree and sentence bracketed with Word Sense

In the last experiment, we combine all aspects that have been using with important word and sentiment word score to determine sentiment polarity. In this experiment, we define a list important words that have a strong effect on the meaning of the whole sentence. Also, based on VietsentiwordNet constructed by Son. X .V (2011). This sentiwordNet contain 1000 word with different positivity and negativity scores. Those scores are ranged from 0 to 1. The objectivity score is calculated by the formula.

$$\text{ObjScore} = 1 - (\text{PosScore} + \text{NegScore})$$

Where ObjScore is objectivity score, PosScore is positivity score, and NegScore is negativity score. The accuracy was improved by 3 percent from 86.04% to 89.6% in comparison with tree and sentence bracketed with rewarded and reversed voting without the use of word sense.

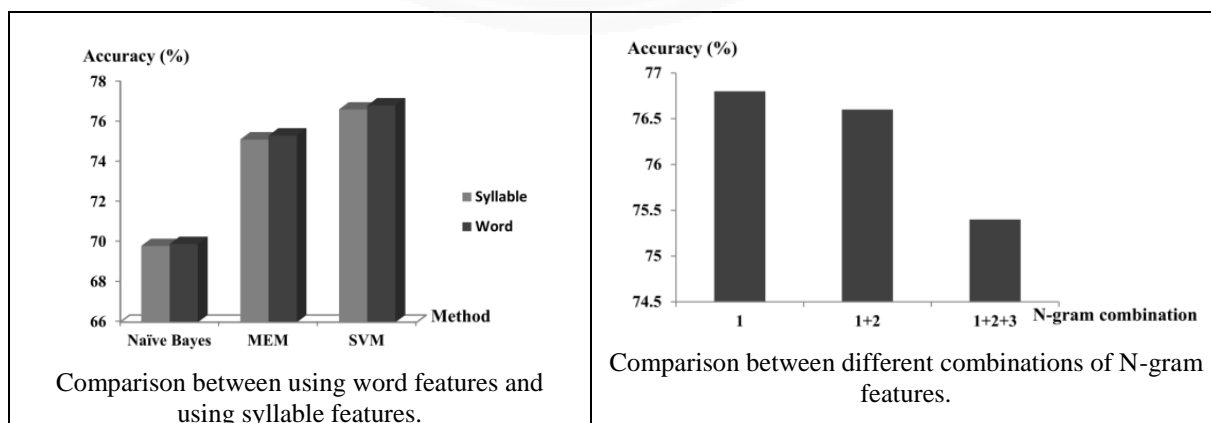
4.2.4 Overall evaluation.

The best accuracy result is achieved by dependency tree with consideration of word sense. Also, this method has the highest measurement by Recall. Application of dependency with *reversed* and *rewarded* relation is performed better than the application of dependency tree with belief propagation. The highest performance by Precision measurement (90.59%) is produced by dependency tree with belief propagation. However, we regarded it is not an accurate measurement because the belief propagation performed pretty well on the classification of the positive sentence but its classification process ran poorly on negative sentences. Therefore, we have more number of true positive (t_p) than false positive (f_p). In turn, by calculation of formula (4), we gained high measurement of Precision.

In overall, three proposed methods performed better than feature selection with machine learning techniques. For instance, the best measurement from our method is 90.63% by Recall, while the best measurement from feature selection technique is 69.3% by Recall. This comparison shows a superior performance of our proposed method on other technique. We believe that our methods prove a strong improvement on sentiment classification since it regarded the sentence as a dependency graph. Moreover, it treats each component in a sentence as a meaningful phrase rather individual words.

4.3 Comparison with other experiments

In order to evaluate the performance of our methods, we make a comparison between our archived results with an experiment done by Duyen .N .T (2014). Both experiments were conducting on the same data set, but we used a different technique. In Duyen’s research, Naïve Bayes, MEM, and SVM are three main methods were used. For each learning method, they conducted their experiment with a combination of different features such as word features, syllable features, important word features, important syllable feature, n-gram features, and overall score features. Four graphs below show the performance of selecting methods and features.



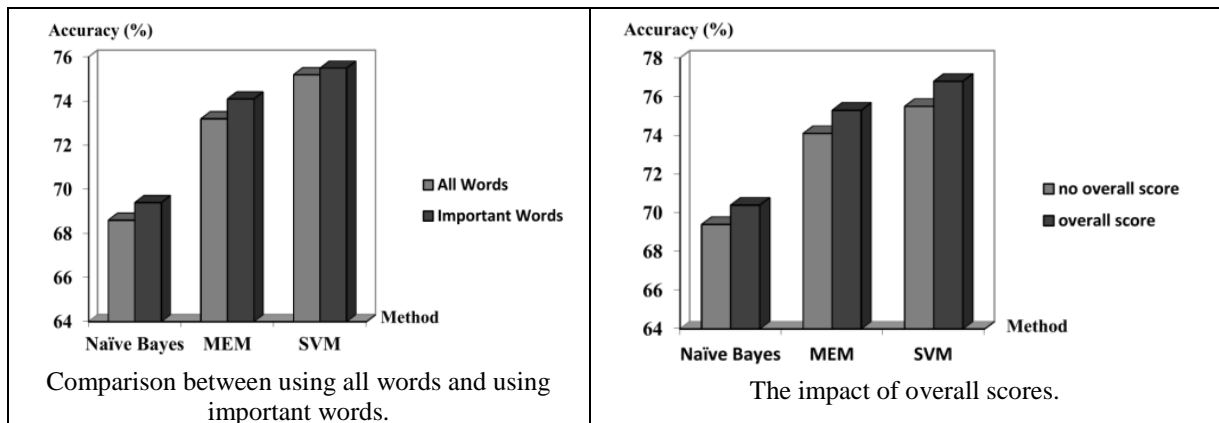


Figure 4.4 Result of Sentiment Analysis for Vietnamese by Duyen .N .T (2014).

As we could observe from above graph, the best result was delivered by a combination of SVM and use of overall score. When the author combines overall scores, the system archived 74.5%, 74.1%, and 69.4% accuracy for SVM, MEM, and Naïve Bayes respectively. Those results are slightly better than our combination of terms features with machine learning technique in accuracy measurement. However, in some other measurement categories, our techniques perform better. For instance, a combination of Naïve Bayes with feature selection can gain the result of 91.8% by Recall measurement for Positive class. This result is much higher than 80.8% result achieved by Duyen .N .T for the same machine learning technique in table 4.3. Generally, reference from Table 4.1, a combination of term feature selection and machine learning measurement for Negative class is still less efficiency than Duyen .N .T work. This can be explained by a reason that data set of the Positive class are greater than Negative class. We can easily determine the positivity of a sentence, but it is more difficult to conclude whether a sentence is negative or somewhat negative. This reason shows that sentence dependency tree is helpful to remove this ambiguity.

Method	Class	Precision	Recall	F ₁
Naive Bayes	Positive	85.4	80.8	83.0
	Negative	66.3	62.4	64.3
MEM	Positive	88.7	90.7	89.7
	Negative	65.5	89.2	75.5
SVM	Positive	82.2	94.0	87.7
	Negative	73.7	72.6	73.1

Table 4.3: Performance of the system on Positive and Negative classes (Duyen .N .T, 2014)

A big difference comes when we compare our dependency tree method with Duyen .N .T research. The highest result we can achieve was delivered by the implementation of dependency tree with word relationship and word sense. The best accuracy was 89.60% while the best accuracy done by Duyen's work was 76.8% with an overall score.

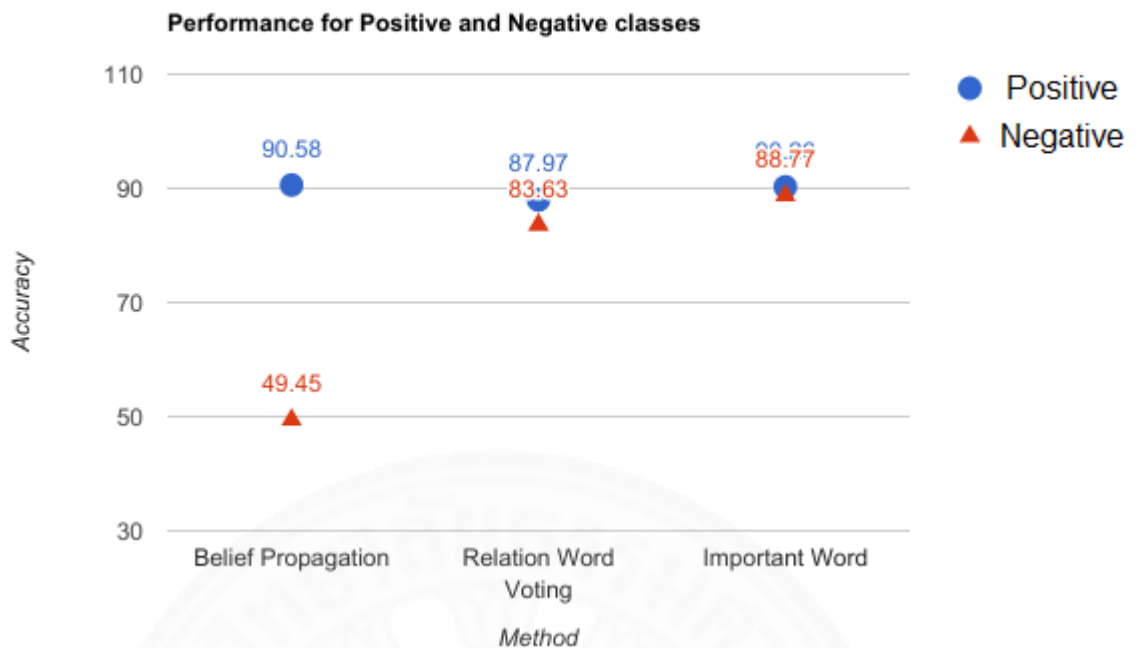


Figure 4.5: Performance for each class

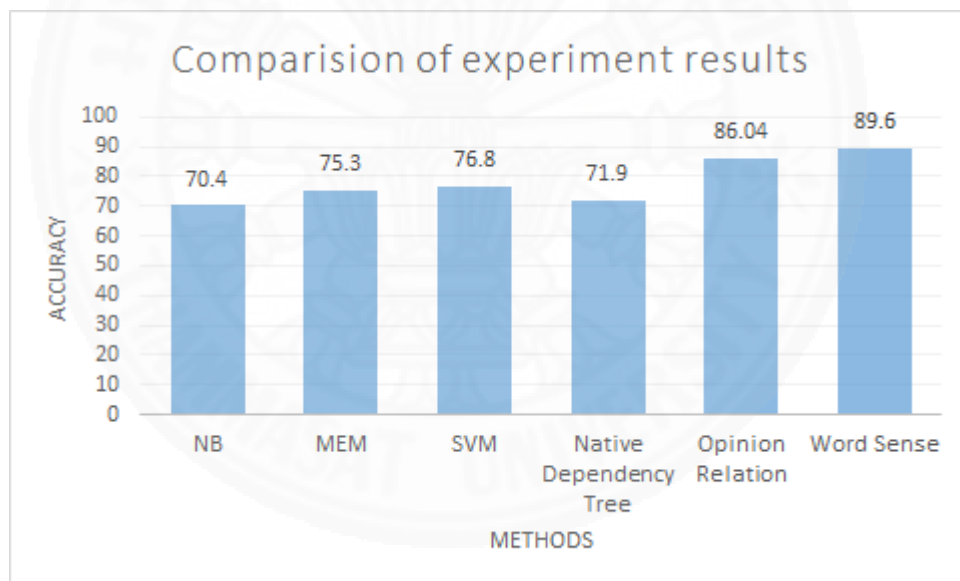


Figure 4.6: Comparison with the previous experiment

In overall, three proposed methods performed better than normal machine learning techniques in N. T. Duyen [30] research. In figure 10 we see that a Supported Vector Machine (SVM) has an accuracy of 76.8% while dependency tree with the application of word sense can archive 89.6%. The best measurement from our method is 90.63% by Recall, while the best classification result for the positive class is 90.58% by dependency tree with belief propagation (Fig. 11). This comparison shows a higher achievement of our proposed method over other techniques. We believe that our methods prove a strong improvement on sentiment classification since it regarded the sentence as a dependency graph. Moreover, it treat each component in a sentence as a meaningful phrase rather individual words

Figure 4.5: Accuracy result for Negative and Positive classes by dependency tree technique. We gained a very high performance in classification of positive class in term of accuracy. The highest recorded result was achieved by combination of dependency tree and belief propagation technique with an accuracy of 90.58%. Other techniques performance were slightly lower than the first one. The reason that combination of dependency tree and belief propagation has the highest score is because it is quite straightforward implementation, and most of the case it does not have to deal with *reversed* and *rewarded* relation. However, when we have to deal with those relations in negative class, this technique showed its weakness. The accuracy result felt down to 49.45% which is very low result in comparison with other technique. If we include *reversed* and *rewarded* relation for implementing our technique, the result increases up to 83.63%. These scenarios show that it is essential that word relation is very helpful to improve sentiment classification.

4.4 Summary

This chapter presents results that we can archive by methodologies presented in section 3. In general, a combination of term feature selection and machine learning technique gained fairly good result in compare with a currently available experiment done by another researcher for the Vietnamese language. Amazingly, sentence dependency tree technique delivers better performance

Chapter 5

Conclusion and Future Work

In this study, we have presented a Vietnamese sentiment analysis based on a combination of machine learning, feature selection, and sentiment tree structure. The experiment was conducting on a corpus extracted from Vietnamese hotel reviews. The experiment result shows that sentiment classification based on sentence representation of tree structure has better accuracy than the combination of sentiment features and machine learning techniques. However, our method still has some limitation such as it is difficult to determine word meaning based on context. This limitation leads to fault rejection in negative class. Choosing the right meaning of the word indifference context is the key to improving the performance of text classification. In next chapter, we will suggest some further techniques that we can implement in future research.

Beside the above learning models, there are a number of advanced methods that utilized the external information to boost the performance of parsing systems to higher levels. Socher et al. (2013) used the deep learning technique, which was based on the recurrent neural network and reaches the F-score of 90.5%. Charniak and Johnson (2005) proposed a general framework called Re-ranking parser. This framework first used a baseline generative parser (such as one in Collins (1999) or in Petrov and Klein (2007)) to produce top k-best candidate parse trees and then used a discriminative model with a set of strong and rich features to re-rank them and pick out the best one. This work used maximum entropy model as a discriminative re-ranker for the baseline system, which could achieve a high F-score of 91.5% on a test set of English Treebank. Huang (2008) improved the strategy for the re-ranking parsers that could encode more candidate parse trees in the first phase and utilize the averaged perceptron model to perform the re-ranking phase, reaching up to F-score of 91.8% on English test set. However that is not the whole story, McClosky et al. (2006) even extended the idea of re-ranking parser by injecting more unsupervised features from a large external text corpus, making the parser become a self-trained system that could achieve a F-score of 92.4% on the test set. Currently, the self-trained parser has been considered as the state-of-the-art parsers in terms of F-score on the English test set.

Also, deep learning model for natural language processing is a promising technology since it has proved itself in sentiment classification. It has the capability of processing a large amount of data with very high accuracy. The learning model has many layers and hidden layers for better language modeling

References

1. Andrea Esuli and Fabrizio Sebastiani, *SentiWordNet: A High-Coverage Lexical Resource for Opinion Mining*. Kluwer Academic Publishers, 2006.
2. Subhabrata Mukherjee, Pushpak Bhattacharyya. *Feature Specific Sentiment Analysis for Product Reviews*. Computational Linguistics and Intelligent Text Processing Volume 7181 of the series Lecture Notes in Computer Science pp 475-487, 2012.
3. K. Dave, S. Lawrence and D.M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. *Proceedings of 12th International Conference on World Wide Web*, pages 519-528. 2003..
4. Diep, B. Q. (2005). *Vietnamese grammar*. Vietnam Education Publisher.
5. M. Taboada. 2006. *Discourse markers as signals (or not) of rhetorical relations*. In *Journal of Pragmatics* 38, pp. 567–592.
6. Tetsuji Nakagawa*, Kentaro Inui. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 786–794, Los Angeles, California, June 2010.
7. Jeevanandam Jotheeswaran* and S. Koteeswaran. *Feature Selection using Random Forest method for Sentiment Analysis*. *Indian Journal of Science and Technology*, Vol 9(3), DOI: 10.17485/ijst/2016/v9i3/75971, January, 2016.
8. Subhabrata Mukherjee, Pushpak Bhattacharyya. *Feature Specific Sentiment Analysis for Product Reviews*. Computational Linguistics and Intelligent Text Processing Volume 7181 of the series Lecture Notes in Computer Science pp 475-487, 2012.
9. Morgane Marchand, Alexandru-Lucian Ginsca. *[LVIC-LIMSI]: Using Syntactic Features and Multi-polarity Words for Sentiment Analysis in Twitter*. Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 418–424, Atlanta, Georgia, June 14-15, 2013.
10. Daniel Ansari. *Sentiment Polarity Classification using Structural Features*. 2015 IEEE 15th International Conference on Data Mining Workshops, pages 1270 – 1273, 14-17 Nov. 2015.
11. M. Porter. 1980. *An algorithm for suffix stripping*. In *Program*, Vol. 14, no. 3, pp. 130–137.
12. Duyen N. T. *An Empirical Study on Sentiment Analysis for Vietnamese*. The 2014 International Conference on Advanced Technologies for Communications (ATC'14), 2014.
13. Dominique Ziegelmayer. *Sentiment polarity classification using statistical data compression models*. 2012 IEEE 12th International Conference on Data Mining Workshops 978-0-7695-4925-5/12, 2012.
14. S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
15. Kieu B. T. and Pham S. B. *Sentiment Analysis for Vietnamese*. 2010 Second International Conference on Knowledge and Systems Engineering. Page 152 – 157, 2010.

16. Ngo Xuan Bach, Pham Duc Van, Nguyen Dinh Tai, Tu Minh Phuong. *Mining Vietnamese Comparative Sentences for Sentiment Analysis*. 2015 Seventh International Conference on Knowledge and Systems Engineering. 978-1-4673-8013-3/15 \$31.00 © 2015 IEEE DOI 10.1109/KSE.2015.36. Pages 162 – 167, 2015.
17. Peifeng Li, Qiaoming Zhu. *A Dependency Tree based Approach for Sentence-level Sentiment Classification*. 2011 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. Pages 166 – 171, 2011.
18. Tetsuji Nakagawa. *Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables*. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 786–794, Los Angeles, California, June 2010.
19. Bo Pang and Lillian Lee. 2005. *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*. Proceedings of the main conference of ACL.
20. Ryan McDonald. *Multilingual dependency analysis with a two-stage discriminative parser*. CoNLL-X '06 Proceedings of the Tenth Conference on Computational Natural Language Learning Pages 216-220, 2006.
21. S. Buchholz, E. Marsi, A. Dubey, and Y. Krymolowski. 2006. *CoNLL-X shared task on multilingual dependency parsing*. SIGNLL.
22. Hong Phuong Le, Tuong Vinh Ho. *A Maximum Entropy Approach to Sentence Boundary Detection of Vietnamese Texts*. IEEE International Conference on Research, Innovation and Vision for the Future - RIVF 2008, Jul 2008, Ho Chi Minh City, Vietnam. 2008.
23. NGUYEN P. T., XUAN L. V., NGUYEN T. M. H., NGUYEN V. H. & LE-HONG P. (2009). Building a large syntactically-annotated corpus of Vietnamese. In *Proceedings of the 3rd Linguistic Annotation Workshop, ACL-IJCNLP*, Singapore.
24. Phuong Le-Hong, Azim Roussanly, Thi Minh Huyen Nguyen, Mathias Rossignol. *An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts*. TALN 2010, Montréal, 19–23 juillet 2010.
25. Hong Phuong Le, Thi Minh Huyen Nguyen, Azim Roussanly, Tuong Vinh Ho. *A Hybrid Approach to Word Segmentation of Vietnamese Texts*. 2nd International Conference on Language and Automata Theory and Applications - LATA 2008, Mar 2008, Tarragona, Spain. Springer Berlin / Heidelberg, 5196, pp.240-249, 2008, Lecture Notes in Computer Science; Language and Automata Theory and Applications.
26. ISO/TC 37/SC 4 AWI N309, Language Resource Management - Word Segmentation of Written Texts for Mono-lingual and Multi-lingual Information Processing- Part I: General Principles and Methods. Technical Report, ISO, 2006.
27. [2] F. Sebastiani. *Machine Learning in Automated Text Categorization ACM Computing Survey*, 34(1): 1 - 47, 2002.
28. Y. Yang and X. Liu. *A re-examination of text categorization*. In Proc. of the 22nd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Morgan Kaufmann, pp. 42-49 (1999).

29. John C. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. Microsoft Research 1 Microsoft Way, Redmond, WA 98052, USA.
30. D. Lewis. Naive bayes at forty: *The independence assumption in information retrieval*. Proc. of European Conf. on Machine Learning, pages 4–15, 1998.
31. Jay Gholap. “*Performance tuning of j48 algorithm for prediction of soil fertility*”. Dept. of Computer Engineering College of Engineering, Pune, Maharashtra, India.
32. H. Liu, M. Motoda, L. Yu, “*Feature Extraction, Selection, and Construction*”. In N. Ye (eds.): *The Handbook of Data Mining*, Lawrence Erlbaum Associates, Inc. Publishers, pp. 409-423, 2003.
33. F. Sebastiani. *Machine Learning in Automated Text Categorization*. ACM Computing Survey, 2002, 34(1): 1 - 47.
34. Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.
35. George H. John and Pat Langley (1995). *Estimating Continuous Distributions in Bayesian Classifiers*. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338-345. Morgan Kaufmann, San Mateo.
36. A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. In Proc. ACL.
37. Y. Ding and M. Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In Proc. ACL.
38. Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. 2002. Automatic paraphrase acquisition from news articles. In Proc. HLT.
39. Dai Q. N. *From Treebank Conversion to Automatic Dependency Parsing for Vietnamese*. Natural Language Processing and Information Systems Volume 8455 of the series Lecture Notes in Computer Science. pp. 196-207, 2014.
40. David J. C. MacKay. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.
41. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. *Thumbs up? Sentiment Classification using Machine Learning Techniques*. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, pp 79–86.
42. B. Pang, L. Lee, “*Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*,” Proceedings of the ACL, 2005.
43. Yan LI, L. Zhen QIN, “*Unsupervised Sentiment-Bearing Feature Selection for Document-Level Sentiment Classification*,” IEICE TRANS. INF. & SYST., VOL.E96-D, NO.12 DECEMBER 2013.
44. Olena Kummer, “*Feature Selection in Sentiment Analysis*,” CORIA 2012, pp. 273–284, Bordeaux, 21-23, March 2012.
45. Daniel Ansari, “*Sentiment Polarity Classification using Structural Features*,” 2015 IEEE 15th International Conference on Data Mining Workshops, pp 1270-1273, 2015.
46. D. Lewis. *Naive bayes at forty: The independence assumption in information retrieval*. Proc. of European Conf. on Machine Learning, pages 4–15, 1998.
47. John C. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. Microsoft Research 1 Microsoft Way, Redmond, WA 98052, USA.
48. M. Taboada. *Discourse markers as signals (or not) of rhetorical relations*. In Journal of Pragmatics 38, pp. 567–592, 2006.
49. W. C. Mann and S. A. Thompson. *Rhetorical Structure Theory: Toward a functional theory of text organization*. In Text, 8 (3), 243–281, 1988.

50. Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan- Kauffman. 2002.
51. Ugan Yasavur, Jorge Traviesso. *Sentiment Analysis Using Dependency Trees and Named-Entities*. Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference, pp 134-139, 2014.
52. David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
53. R. Hudson. *Word Grammar*. Blackwell, 1984.
54. I.A. Mel'cuk. *Dependency Syntax: Theory and Practice*. State University of New York Press, 1988.
55. Y. Ding and M. Palmer. *Machine translation using probabilistic synchronous dependency insertion grammars*. In Proc. ACL, 2005.
56. R. McDonald. *Discriminative sentence compression with soft syntactic constraints*. In Proc. EACL, 2006.
57. A. Haghighi, A. Ng, and C. Manning. *Robust textual inference via graph matching*. In Proc. HTLEMNLP, 2005.
58. T. Wilson, J. Wiebe, and P. Hoffmann. *Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis*. Computational Linguistics, vol. 35, no. 3, pp. 399-433, 2009.
59. A. Go, R. Bhayani, and L. Huang. *Twitter Sentiment Classification using Distant Supervision*. CS224N Project Report, Stanford 1, 12.
60. S.Arora, E. Mayfield, C. Penstein-Ros'e and E. Nyberg. *Sentiment Classification using Automatically Extracted Subgraph Features*. CAAGET '10 Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pp. 131-139, 2010.
61. W. Zhang, Q. Zhu, and P. Li. *Sentiment Classification Based On Syntax Tree Pruning and Tree Kernel*. Proc. WISA 2010, pp. 101- 105, 2010.
62. H. L. Hammer, P. E. Solberg, and L. Øvrelid *Sentiment classification of online political discussions: a comparison of a word-based and dependency-based method*. Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 90–96, Baltimore, Maryland, USA. June 27, 2014.
63. D. Ziegelmayr and R. Schrader. *Sentiment polarity classification using statistical data compression models*. IEEE 12th International Conference on Data Mining Workshops, pp. 731-738, 2012.
64. T. Nakagawa, K. Inui and S. Kurohashi. *Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables*. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pp 786–794, Los Angeles, California, June 2010.
65. P. Li, Q. Zhu, W. Zhang. *A Dependency Tree based Approach for Sentence-level Sentiment Classification*. 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 166-171, 2011.
66. C. Quan, X. Wei, F. Ren. *Combine Sentiment Lexicon and Dependency Parsing for Sentiment Classification*. Proceedings of the 2013 IEEE/SICE International Symposium on System Integration, pp. 100-104, Kobe International Conference Center, Kobe, Japan, December 15-17, 2013.
67. B. Li, L. Zhou, S. Feng and K. Wong, "A Unified Graph Model for Sentence-based Opinion Retrieval," Proc. ACL 2010, pp. 1367–1375, 2010.

68. V. Ng, S. Dasgupta, S. Niaz Arifin, "Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews", Proc. ACL 2006, pp. 611-618, 2006.
69. S. Matsumoto, H. Takamura and M. Okumura, "Sentiment Classification using Word Sub-sequences and Dependency Subtrees," Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 301-310, 2005.
70. K. Dave, S. Lawrence, and D. M. Pennock. *Mining the peanut gallery: Opinion extraction and semantic classification of product reviews*. In Proc. of WWW, pages 519-528, 2003.
71. J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. *Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques*. In Proc. of the IEEE International Conference on Data Mining (ICDM), 2003.
72. Bang, T. S., Choochart H., and Virach S. *Vietnamese sentiment analysis based on term feature selection approach*. In Proceedings of the 10th International Conference on Knowledge Information and Creativity Support Systems (KICSS 2015) [CD-ROM], 12-14 November 2015, Phuket, Thailand, pp. 196-204, 2015.
73. Duyen N.T. *An Empirical Study on Sentiment Analysis for Vietnamese*. The 2014 International Conference on Advanced Technologies for Communications. 2014.
74. Dai Q. N. *From Treebank Conversion to Automatic Dependency Parsing for Vietnamese*. Natural Language Processing and Information Systems Volume 8455 of the series Lecture Notes in Computer Science pp 196-207.
75. David J. C. MacKay. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.
76. J. S. Yedidia, W. T. Freeman, and Y. Weiss. 2004. *Constructing free-energy approximations and generalized belief approximation algorithms*. MERL TR2004-040, Mitsubishi Electric Research Laboratories.
77. Braunstein, A., Mézard, R., Zecchina, R. "Survey propagation: An algorithm for satisfiability". *Random Structures & Algorithms*. 27(2): 201–226. doi:10.1002/rsa.20057, 2005.
78. Vu XS., Song HJ., Park SB. (2014) *Building a Vietnamese SentiWordNet Using Vietnamese Electronic Dictionary and String Kernel*. Knowledge Management and Acquisition for Smart Systems and Services Volume 8863 of the series Lecture Notes in Computer Science pp 223-235.
79. Kroeger Paul. *Analyzing Grammar: An Introduction*. Cambridge University Press, May 2005. ISBN 9780521816229