PAPER *Special Section on Intelligent Information and Communication Technology and its Applications to Creative Activity Support*

Sentiment Classification for Hotel Booking Review Based on Sentence Dependency Structure and Sub-Opinion Analysis

Tran Sy BANG[†a)] ***and*** **Virach SORNLERTLAMVANICH**[†b)], ***Members***

SUMMARY This paper presents a supervised method to classify a document at the sub-sentence level. Traditionally, sentiment analysis often classifies sentence polarity based on word features, syllable features, or N-gram features. A sentence, as a whole, may contain several phrases and words which carry their own specific sentiment. However, classifying a sentence based on phrases and words can sometimes be incoherent because they are ungrammatically formed. In order to overcome this problem, we need to arrange words and phrase in a dependency form to capture their semantic scope of sentiment. Thus, we transform a sentence into a dependency tree structure. A dependency tree is composed of subtrees, and each subtree allocates words and syllables in a grammatical order. Moreover, a sentence dependency tree structure can mitigate word sense ambiguity or solve the inherent polysemy of words by determining their word sense. In our experiment, we provide the details of the proposed subtree polarity classification for sub-opinion analysis. To conclude our discussion, we also elaborate on the effectiveness of the analysis result.

key words: *sentiment analysis, sentence dependency parsing, subtree opinions, Vietnamese sentiment classification, hotel review classification*

1. Introduction

An online hotel booking service is an indispensable for travellers and international backpackers. Hotel owners extend their business by developing booking services through online platforms such as websites, blogs, or social media, etc. Availability of online booking platforms creates a high volume of feedback data which are continuously delivered to service providers. In order to deal with many customer opinions, automatic sentiment analysis is vitally needed. The typical approach uses a supervised method which requires a big data set for training [1], and machine learning techniques (based on bag-of-words, bi-gram, N-gram, etc.) [2], and feature selections [3]–[5]. Those methods focus on words or groups of words that occur in a corpus and use common learning algorithms like Naïve Bayes (NB) [6] or Support Vector Machine (SVM) [7] for classification. Sentences usually contain multiple phrases which are linked to each other by meaningful conjunction words like *but, although, however*, etc. Rhetorical Structure Theory stated that 90% of rhetorical relations is triggered by connective words in articles [8], [9]. Dependency tree-based sentiment classification is proposed by introducing CRF with hidden variables of dependent subtree polarity to classify the sentence sentiment [10]. The approach is similarly used to solve the problem of subtree polarity reversal, but without considering the word sense.

Traditional domain-oriented text classification commonly treats word elements of a sentence separately. For instance, when the majority of a specific category word is likely to occur in a sentence, this sentence will be highly classified into a similar category. Furthermore, a sentence consists of several sub-opinions, and each sub-opinion could interact with each other for delivering a total sentence opinion. For example, we select a sentence from our collected corpus "*The new glass blocks noise and heat from outside*". Generally, "*noise*" and "*heat*" are negative words, therefore the sub-opinion of "*noise and heat*" is most likely a high level of negativity. Additionally, "*blocks*" has a meaning "to reverse" the sense of its succeeding words, "*noise and heat*". As a result, the total sentence polarity is changed into positive. Considering another example of "*The hotel is far from the city, but service is excellent*", the sub-opinion of "*far from city*" has negative polarity because the word "*far*" is considered to contain a negative meaning. Meanwhile, a sub-opinion of "*service is excellent*" has positive polarity because the word "*excellent*" is a positive word. The first opinion is generally neglected in general dialog of the Vietnamese language because a speaker wants to emphasize the good service offered by the hotel. This is because the word "*but*" is indicated to contrast the relation between the sub-opinions. Finally, the total polarity of the sentence is decided by its succeeding sub-opinion.

Accordingly, the parsing model that generates a dependency graph representation of a sentence can hold the information of the polar words and their relations. Words and their arguments can be modeled through directed edges, leaves, and nodes [11], [12]. Also, a dependency graph contains rich features that can be further used for language processing. Those features are included in machine translation [13], sentence compression [14], and textual inference [15].

Besides, another important aspect of a sentence dependency structure is a grammatical relation. A sentence, as a whole, can be ambiguous, although its sub-opinions are unambiguous. For instance, the sentence "*customer can listen to nice room*" is ambiguous because it is uncommon in daily conversation. A dependency structure can express a sentence in a grammatical way by assigning a grammatical relation within a clause, as direct and indirect arguments. This

Manuscript received March 17, 2017.
Manuscript revised July 31, 2017.
Manuscript publicized January 19, 2018.

[†]The authors are with SIIT, Thammasat University, 12120 Thailand.
a) E-mail: bangtran365@gmail.com (Corresponding author)
b) E-mail: virach@gmail.com

DOI: 10.1587/transinf.2016IIP0038

includes subject, primary object, and secondary object. Because arguments are closely related to each other in meaning a dependency tree plays an important role in determining a suitable word in a sentence.

In this paper, we conduct our experiment on a corpus collected from a popular hotel booking site Agoda[†]. We devise a supervised method that used a sentence dependency structure and rule-based system that provides linguistic compositionality rules. In this method, the polarity of the whole sentence is regarded as the sentence level opinion, and the polarity of the component terms are regarded as the sub-opinions. The polarity of the whole sentence reflects the sentence level, and it is calculated under consideration of the interaction between the polarity of the sub-opinions. This study aims to improve sentence-level classification, which leads to the following:

- Classify a sentence polarity based on sub-opinions in the dependency tree and sum-product belief propagation.
- Classify a sentence polarity based on sub-opinions in the dependency tree and sub-opinion relations.
- Classify a sentence polarity based on the sentence dependency tree, sub-opinions, and word granularity.

The rest of the paper is organized as the following. Section 2 presents related works that use dependency parsing tree for sentiment analysis. Section 3 describes the method of using dependency parsing tree with sub-opinions and sub-opinions relations. Section 4 shows experiment results, and Sect. 5 concludes the results.

2. Related Works

Sentiment classification attracts the attention of many researchers, and various kinds of research have been implemented in different aspects i.e. word aspect [15], sentence aspects [16]–[19], and document aspects [20]. This research focuses on sentence level classification based on sentence structure. Therefore, the review of the related works is evolved with sentence dependency structure and its subtrees.

Li [20] mined dependency structure features in combination with flat features from the dependency tree to form a novel feature for a sentence. The structure features together with flat features are applied to a convolution tree kernel-based approach for sentence classification with dependency parse trees, and current polarity of named-entities based on the fly topic modeling. The performance of the system was shown by a comparison of three rule-based approaches and two supervised approaches (i.e. Naive Bayes and Maximum Entropy).

Quan [21] proposed a method of combination of lexicon-base and dependency parsing for sentiment classification. Quan [21] selected a relationship between an adverb and target word by using a dependency parser. A word dictionary is constructed based on the HowNet similarity method. A summation of the weighted sentiment of all phrases is calculated for determination of the final sentence polarity. Li et al. [22] used structure feature of topic and sentiment word pairs to represent the contextual information of opinions and target topics. Also, they used this contextual information to detect opinions toward the same topic. From two kinds of relations, the author constructed a graph-based model for opinion mining. Another approach is from Matsumoto [23], who mined the most frequent dependency sub-tree that was derived from corpus sentences. The most occurring sub-trees were recorded as new features for sentiment classification. Dave et al. [24] explored the use adjective-noun relations, for a noun with relative dependency polarity. Nakagawa [10] applied the dependency structure and treated each node in the dependency tree, except the root note, as a hidden variable. The approach classifies the sentence by applying the Conditional Random Field (CRF) method on the hidden variables. None of the surveyed papers uses *sub-opinions relation* and *word granularity*, which are expected to capture a better opinion granularity. This paper improves the sentiment classification by using a corpus of Vietnamese hotel booking opinions. The approach uses the text domain. In this case, we conducted the experiment on the Vietnam hotel reviews and confirmed the performance of our approach. It is applicable to other languages and domains but needs an annotated corpus and appropriate analysis support from the WordNet for the specific terms.

3. Methodology

3.1 Sentiment Classification System Based on Dependency Sub-Trees

Let us consider the subjective sentence "*The new window prevents mosquitos and flies but the swimming pool is very crowded*" (indoor pool). Its dependency tree is shown in Fig. 1, and its sub-opinion, corresponding to each dependency subtree, is shown in Fig. 2. The polarity sign (+ and −) is labelled in the circle at the head of the dependency

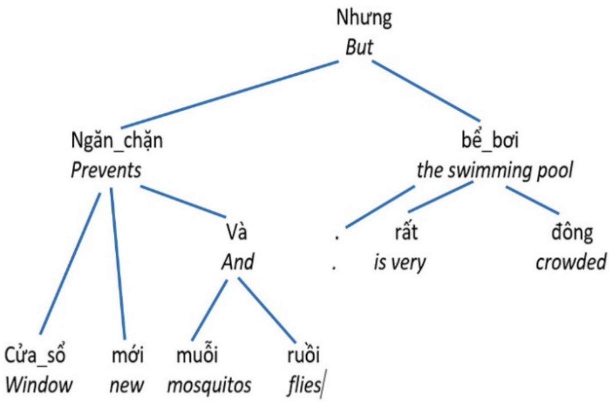

Fig. 1 Dependency tree of the complete sentence.

[†] https://www.agoda.com/

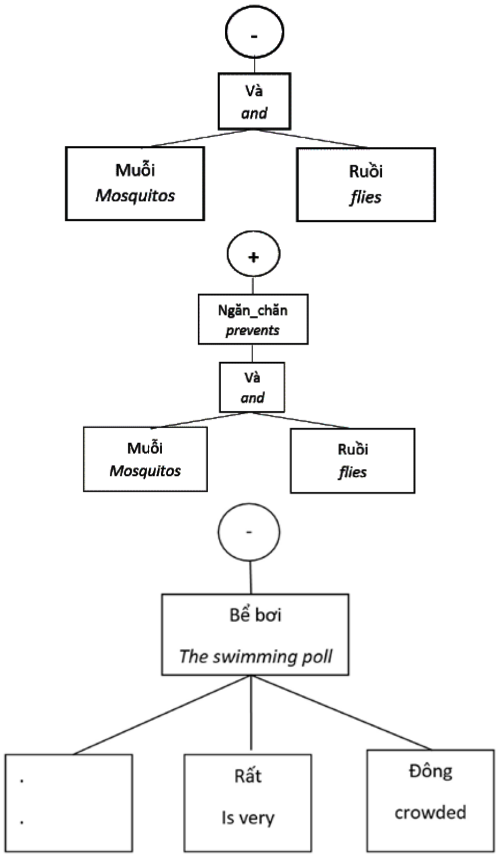

Fig. 2 Polarities of dependency sub-trees.

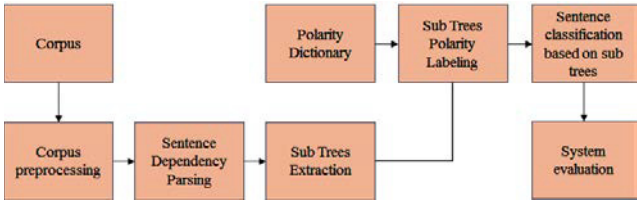
Corpus
Polarity Dictionary
Sub Trees Polarity Labeling
Sentence classification based on sub trees
Corpus preprocessing
Sentence Dependency Parsing
Sub Trees Extraction
System evaluation

Fig. 3 System overview.

structure. In this sentence "*mosquitos*" and "*flies*" are considered as negative polarities in the hotel review domain. In sentiment classification that relies on N-gram, word feature selection [25] could wrongly classify the phrase "*The new window prevents mosquitos and flies*" as a negative polarity because the polarities of two words "*mosquitos*" and "*flies*" are reversed by modifying the word "*prevents*". In addition, in a sentence, a conjunction word "*but*" joins the first phrase "*prevents mosquitos and flies*", and the second phrase "*the swimming pool is very crowded*". This conjunction word is considered as a rewarded word that emphasizes the sentence polarity in the second phrase. In this manner, we can determine the sentence polarity based on dependency subtrees of a subjective sentence rather than considering each individual word because the dependency structure of word phrases provides a better scope of the polarity.

Figure 3 describes the whole system consisting of following components:

```
<review Score="8" id="4">
      <sentence   Class="POSITIVE"   Id="1">Nhân_viên/N
khách_sạn/N  rất/R  thân_thiện/A  ,/,  hòa_đồng/V  ./.
</sentence>
   </review>
```

Translation:*The staffs are very friendly and hospitable*

Fig. 4 Corpus format

Table 1 Accuracy results (%) on gold standard POS tags.

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

Corpus contains sentence ID and overall score for the comments. The score is given based on hotel conditions/cleaniness, location, and value for money, facilities, and service. In order to evaluate the efficiency of our method, we use a corpus collected from Agoda, an online booking website for tourist attractions in Viet Nam. The comments in review boxes consist of many languages, but only the Vietnamese language review boxes are considered in this study. The corpus is partially adapted from Duyen [26], and it is introduced to balance the number of negative and positive sentences. In total, we collected 4,011 review sentences from approximately 300 hotels. The sentence polarities are labeled by two annotators. In order to measure the inter-annotator agreement, we use the Cohen's kappa coefficient as follows:

$$K = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{1}$$

where $\Pr(a)$ is the relative observed agreement between two annotators, and $\Pr(e)$ is the hypothetical probability of chance agreement. The Cohen's kappa coefficient of our corpus was 0.89, which can be interpreted as almost perfect agreement.

Corpus processing uses **vnTokenizer**[†] to segment the sentences. This software is designed for tokenizing Vietnamese texts. It segments Vietnamese texts into lexical units (words, names, dates, numbers and other regular expressions) with a high accuracy, of about 98% on the test set extracted from the Vietnamese Treebank.

From corpus collection observation, we notice that most of the review sentences are short and less than 30 words. In addition, the reviewers often break a sentence without considering grammatical rules. Therefore, it is suitable to deploy MSTParser to construct dependency trees since it performs well on short sentences.

The Sentence dependency parsing model is constructed, based on Vietnamese dependency Treebank VnDT, which contains 10,200 sentences. The dependency trees outputs are represented in the form of the CoNLL 10-column

[†] http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTokenize

standard. The output contains node features and edge features, the root of the tree, and tagged words. We apply the vnDP model developed from Dai [27] to handle this task. From the CoNLL format, we successfully bracket the phrases in the sentence so that we can extract them to be a subtree feature. The dependency tree is presented in a hierarchical dependency graph. It has a big advantage in visualizing tree structure but it is difficult for the computer reading data process. We decided to convert the tree dependency graph into a sentence bracketed form, so that a computer could easily read sub-tree inputs sequentially. We successfully developed an algorithm to bracket the dependency structure, as shown in Algorithm 1.

Algorithm 1 Sentence Bracketing

```
Input: parent: list of parenthesis character parents=['(', ' )']
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, parents)
    childstrs= “” # start with an empty string
    for a child in self
        if child is instance in Tree:
            childstrs.append(chil.sentence_bracket(parents))
        else if child is instance in tuple:
            childstrs.append(“/”.join(child))
        else
            childstrs.append(child)
    return parents[0], “ ”.join(childstrs), parents[1]
```

The bracket starts with ROOT since the top node is indexed by 0. As it is a head of the tree, the subsequent sub-tree nodes are bracketed with inner brackets.

The Sub-Tree extraction module processes bracketed sentences and produce a list of sub-trees as outputs. The sub-trees contain parent nodes, children nodes, and dependency relation tags. The statistical number of sub-trees for positive and negative relations is illustrated in Table 2.

Sub-trees polarity labeling handles word polarity tagging by the topic domain dictionary. This dictionary contains a list of words that their meanings are already disambiguated by the annotators. The prior polarity of a phrase $\{+1, -1\}$ is the innate sentiment polarity of a word contained in the phrase, which can be obtained from sentiment polarity dictionaries. The resulting dictionary contains 324 positive expressions and 332 negative expressions. We also construct a relation dictionary which contains 37 *reversed* expressions and 13 *rewarded* expressions.

(*ROOT*,ngăn_chặn cửa_sổ mới (*dob*,muỗi (*coord*,và ruồi)) (*coord*,nhưng (*conj*,cho_phép (*dob*,không_khí trong_lành) (*vmod*,đi qua) .)))

Fig. 5 Dependency tree graph is represented in bracketed form.

Table 2 Corpus volume and extracted features

Class	Number of sentences	Number of extracted subtrees
Positive	2187	16304
Negative	1826	13951

Sentence classification based on the sub-tree module concludes the sentence polarity based on its dependency sub-trees. Section 3.2 describes in detail the classification methodology.

3.2 Classification Methodology

3.2.1 Classification by *Sub-Opinions* and *Dependency Tree Propagation*

MacKay (2003, chapters 16 and 26) [28] presented a theory of belief propagation, which is a generalization of the forward-backward algorithm that is deeply studied in the graphical model literature (Yedidia et al., 2004) [29]. 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 [32]. A graph contains nodes, corresponding to variables V and factors F, the edge connects variables and the factors. The joint mass function is:

$$p(\mathrm{x}) = \prod\nolimits_{a \in F} f_a(x_a) \tag{2}$$

where x_a is the vector of neighboring nodes to the factor node a. The function works by passing a *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 $\mu_{v \to a}$ is passed from v to a and $\mu_{a \to v}$ is passed from a to v. The message is computed differently, based on whether a node is a variable node or factor node.

Figure 6 represents a 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.

Before starting, the graph is oriented by designating

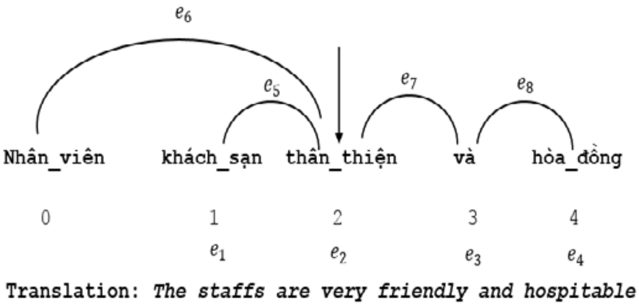

Fig. 6 Node features and edge features in dependency tree.

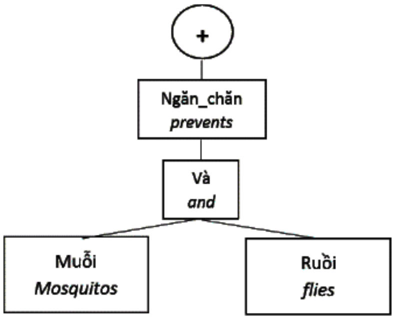

Fig. 7 Subtree contains reversed relation.

one node as the root; any non-root node which is connected to only one other node is called a leaf. In the first step, messages are passed inwards. Starting at the leaves, each node passes a message along the (unique) edge towards the root node. The tree structure guarantees that it is possible to obtain messages from all other adjoining nodes before passing the message on. This continues until the root has obtained messages from all of its adjoining nodes. The second step involves passing the messages back out. Starting at the root, messages are passed in the reverse direction. The algorithm is completed when all leaves have received their messages. Applying this method to the sentence in the Fig. 1, this sentence would be classified as a negative sentence. Since two words "*mosquitos*" and "*flies*" are negative, their polarities are propagated to an upper node and combined with the positive polarity of the word "*fresh*". Since the final sentence polarity is the product of all phrases, the sentence polarity is negative. However, this conclusion is wrong because this method skipped the negation relation word "*prevent*". This weakness will be improved by applying the second method in the next section.

3.2.2 Classification with Consideration of *Sub-Opinion Relation*

When a subtree contains an *opinion relation word* registered in dictionaries, its polarity resulting from calculating from its leaves is reversed. Taking the sentence in the Fig. 1 as an example, since "*prevent*" reverses the polarity of the two words "*mosquitos*" and "*flies*", from negative to positive.

Therefore, the subtree polarity in the Fig. 7 is positive, and by applying the sum product propagation method in Sect. 3.2.1, we can update the polarity of the word "*prevent*" to be positive, which is useful to determine the subtree polarity at a higher level. Similarly, from Fig. 1, we defined "*but*" as a *contradiction meaning word* because it strengthens the polarity of its following phrase. Accordingly, if the sentence has a structure like:

(A but B)

where *A* is the preceding phrase and *B* is the proceeding phrase. Sentence polarity is decided as the polarity of *B*. Algorithm 2 shows a completed procedure for determining a sentence polarity that contains *reversed* and *rewarded* relations.

Algorithm 2 Sentence analysis with *sub-opinion relations*

Input: bracketed sentences form

```
Function
    while has line and line is not empty
        find a pattern m which is a matched parenthesis pair
        while m is found
            sub-opinion= an element in a group of pattern m
            if a group of sub-opinion does not contain sub-opinion
                add new sub-opinion to a group of sub-opinion
    for each sub-opinion in a group of sub-opinion
        if sub-opinion parent node has negation relation
            sub-opinion is reversed
            update parent node polarity
        else if sub-opinion parent node has contrast relation
            sub-opinion polarity is equal to the polarity of its right
            children in its tree structure
            update parent node polarity
        else
            sub-opinion polarity is decided by sum product
            propagation
            update parent node polarity
    return sentence polarity
```

However, we still face one existing issue in that the second phrase "*the swimming pool is very crowded*" obviously has a negative meaning since "*crowded*" is a negative word. But in the Vietnamese language, the word "*đông*" which is translated as "*crowded*" can have several meanings dependings on the different situations. The method in the next section will solve this issue.

3.2.3 Classification with Considering of *Sub-Opinion Relation* and *Word Granularity*

It is a difficult task to determine an appropriate granularity of a word in a different concept. The Vietnamese language does not have a concept of tense for a verb like "*run*" in present tense, which is "*ran*" for past tense. Also, in English, most nouns need to be in a form of singular or plural (e.g. pen vs pens) whereas Vietnamese noun "do not in themselves contain any notion of number or amount" [30]. Taking the second phrase in Fig. 1 as an example, "*bể bơi rất đông*" (the swimming pool is very crowded) and another phrase "*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 is used as an adjective, but in the second sentence, it is used as a noun. Vietnamese is a sort of isolating language in that word formation is a combination of isolated syllables [31]. In the reviews of Vietnamese hotels, the reviewers often use these kinds of homonyms that often lead to the problem of misclassifications. These syntactic aspects are represented by using constituency of syntactic structures. This concept was implemented in the attempt of building Viet Treebank [32]. Viet Treebank consists of a corpus with word segmentation and POS annotation. The vnDT model was constructed based

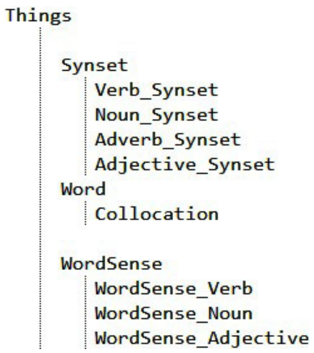

Fig. 8 VietWordNet hierarchical structure

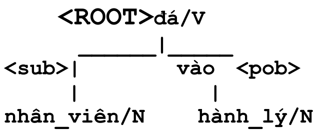

Fig. 9 Example of expressed sentence.

Table 3 Sense for *đá* and *hành lý*

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

on Viet Treebank that can provide the syntactic dependency of word sense by giving its most plausible syntactic analysis [27]. There is a 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 the first sense. In order to utilize word sense with dependency tree, we use Vietnamese Wordnet [33]. Wordnet is an ontology that holds relationships among words in terms of synset (a set of synsonyms), 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 the first sense. The complete set of the first sense granularity is presented below.

Noun first sense: gốc *(origin)*, hành động *(action)*, động vật *(animal)*, vật tạo tác *(artifacts)*, thuộc tính *(attribute)*, cơ thể *(body)*, nhận thức *(awareness)*, truyền thông *(communication)*, sự kiện *(event)*, cảm giác *(feeling)*, thực phẩm *(food)*, nhóm *(group)*, vị trí *(location)*, động cơ *(motive)*, vật *(object)*, tự nhiên *(nature)*, người *(people)*, hiện tượng *(phenomenon)*, thực vật *(plant)*, sở hữu *(possession)*, quá trình *(process)*, lượng *(quantity)*, quan hệ *(relation)*, hình dạng *(shape)*, trạng thái *(state)*, chất *(quality)*, thời gian *(time)*.

Verb first sense: cơ thể *(body)*, biến đổi *(change)*, nhận thức *(awareness)*, truyền thông *(communication)*, thi đấu *(competition)*, tiêu thụ *(consumption)*, tiếp xúc *(contact)*, tạo tác *(creation)*, cảm xúc *(feeling)*, vận động *(motion)*, tri giác *(emotion)*, sở hữu *(possession)*, xã hội *(social)*, trạng thái *(state)*, thời tiết *(weather)*.

Adjective first sense: tính từ (adjective)

We will investigate a sentence:
Nhân viên đá vào hành lý
t*(Staff kicks the luggages)*

This sentence is expressed as in Fig. 9:

A word can hold multiple senses, however, in our research scope, we only address two major senses. In Table 3 we show a possible number of senses for the above sentence.

We assume that the word “*đá(N)#1*” (stone) is related to the first sense “*vật*” (object). Also, the word “*hành lý(N)#1*” (luggage) refers to the non-human thing, and its first sense is strongly related to “*vật*” (object). Thus, a combination of (“*đá(N)#1*” (stone) and, “*hành lý(N)#1*” (luggage)) can be an appropriate sense combination. However, “*Nhân_viên (N) #1*” (staff) refers to a 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 can form many possible sense combinations following the set of senses in Table 3. Finally, we can obtain (*Nhân_viên (N) #1*” (staff), *đá (V) #1* (kick), hành_lý (N) #1 (luggage)) as a suitable sense because *đá (V) #1* (kick) has first sense in a set of actions that effects 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 Sects. 3.2.1 and 3.2.2 will be applied continuously to classify sentence polarity. Similarly, we can figure out that the word “*đông*” has a close meaning to “*crowded*” rather than season “*winter*” because its first sense is related to an object similar to “*swimming pool*”.

4. Experimental Results

In this section, we present the experimental result of our new methods and compare this result with feature selection and machine learning techniques. The results are shown in Table 4, and the standard measures of Recall, Precision, and F-Score are used to evaluate the system performance:

$$Recall = \frac{tp}{tp + fn} \tag{3}$$

$$Precision \ = \frac{tp}{tp + fp} \tag{4}$$

F1-Score is the harmonic mean of both:

$$F1 = \frac{2PR}{P + R} \tag{5}$$

The accuracy of the system is measured by:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{6}$$

Table 4 Experimental result

Methods	Accuracy (%)	Precision (%)	Recall (%)	F (%)
Dependency tree with Sum-production propagation	71.90	**90.59**	68.26	77.85
Dependency tree with opinion relation	86.04	87.97	86.63	87.30
Dependency tree with Word Sense	**89.60**	90.26	**90.63**	**90.45**

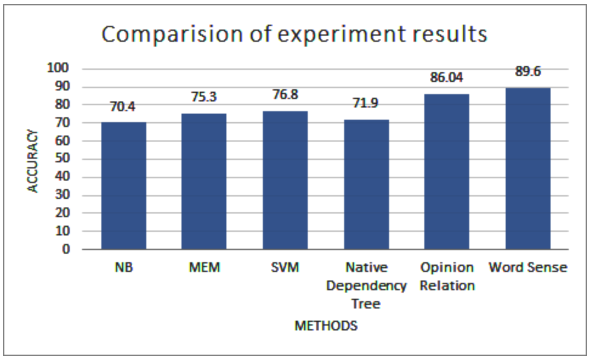

Fig. 10 Comparison with the previous experiment.

where *tp* is True Positive, *tn* is True Negative, *fn* is False Negative, and *fp* is False Positive.

The best accuracy result is achieved by dependency tree with word sense disambiguation. Also, this method has the highest measurement by Recall. Application of dependency with a *sub-opinion 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, this 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 because we have more *true positives (tp)* than *false positives (fp)*. In turn, by calculation of formula (4), we gained better precision.

Duyen has conducted an experiment on the Vietnamese language dataset that we used in this research [27]. In their experiment, the common machine learning techniques are included Naïve Bayes (NB), Maximum Entropy Model (MEM), and Support Vector Machine (SVM). Overall, our three proposed methods performed better than normal machine learning techniques in N.T. Duyen [27] research. In Fig. 10 we see that Supported Vector Machine (SVM) has an accuracy of 76.8% while dependency tree with the application of word sense can achieve 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 better achievement of our proposed method than the other techniques. We believe that our methods are a strong improvement on sentiment classification since it regards a sentence as a dependency graph. Moreover, it treats each component in a sentence as a meaningful phrase, rather than individual words.

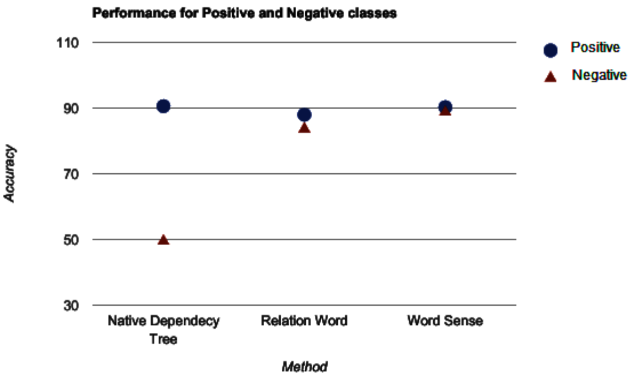

Fig. 11 Classification result for classes

5. Conclusion and Future Work

In this paper, we proposed a new approach for sentiment classification of online hotel booking opinions. We represented sentences in a dependency tree structure, and we mined hidden sub-tree features for conducting classification process. Overall results show a very good performance for feature selection with machine learning techniques.

Despite a good performance, it is essential to further improve the efficiency of our methodologies. One limitation of this research is that word polarity is only determined correctly in a specific domain topic. Low compatibility of the polarity dictionary when we change domain topics is another issue.

Acknowledgments

This work is supported by the Thailand Advanced Institute of Science and Technology; and Tokyo Institute of Technology (TAIST-Tokyo Tech); Sirindhorn International Institute of Technology (SIIT), Thammasat University; and the National Electronics and Computer Technology Center (NECTEC) of Thailand.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” Proc. ACL-02 conference on Empirical methods in natural language processing - EMNLP ’02, pp.79–86, 2002.
[2] B. Pang and L. Lee, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” Proc. 43rd Annual Meeting on Association for Computational Linguistics - ACL ’05, pp.115–124, 2005.
[3] Y. Li, Z. Qin, W. Xu, H. Ji, and J. Guo, “Unsupervised Sentiment-Bearing Feature Selection for Document-Level Sentiment Classification,” IEICE Trans. Inf. & Syst., vol.E96.D, no.12, pp.2805–2813, 2013.
[4] O. Kummer, “Feature Selection in Sentiment Analysis,” CORIA 2012, pp.273–284, Bordeaux, 21-23, March 2012.
[5] D. Ansari, “Sentiment Polarity Classification using Structural Features,” 2015 IEEE 15th International Conference on Data Mining

Workshops, pp.1270–1273, 2015.

[6] D. Lewis, "Naive Bayes at Forty: The Independence Assumption in Information Retrieval." Proc. European Conference on Machine Learning, vol.1398, pp.4–15, Berlin, Heidelberg, 1998.

[7] J.C. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization," Microsoft Research 1 Microsoft Way, Redmond, WA 98052, USA.

[8] M. Taboada, "Discourse markers as signals (or not) of rhetorical relations," J. Pragmatics., vol.38, no.4, pp.567–592, 2006.

[9] W.C. Mann and S.A. Thompson, "Rhetorical Structure Theory: Toward a functional theory of text organization," Text - Interdisciplinary Journal for the Study of Discourse, vol.8, no.3, pp.243–281, 1988.

[10] 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 Association for Computational Linguistics, 2010.

[11] Y. Ding and M. Palmer, "Machine Translation using Probabilistic Synchronous Dependency Insertion Grammars," Proc. 43rd Annual Meeting on Association for Computational Linguistics - ACL '05, pp.541–548, 2005.

[12] R. McDonald, "Discriminative Sentence Compression with Soft Syntactic Constraints," Proc. EACL, 2006.

[13] A.D. Haghighi, A.Y. Ng, and C.D. Manning, "Robust Textual Inference via Graph Matching," Proc. conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05, pp.387–394, 2005.

[14] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis," Comput. Linguist., vol.35, no.3, pp.399–433, 2009.

[15] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," CS224N Project Report, Stanford 1 (12), 2009.

[16] S. Arora, E. Mayfield, C. Penstein-Ros'e, and E. Nyberg, "Sentiment Classification using Automatically Extracted Subgraph Features," CAAGET '10 Proc. NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pp.131–139, 2010.

[17] W. Zhang, P. Li, and Q. Zhu, "Sentiment Classification based on Syntax Tree Pruning and Tree Kernel," Proc. WISA 2010, pp.101–105, 2010.

[18] 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," Proc. 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp.90–96, Baltimore, Maryland, USA, June 27, 2014.

[19] D. Ziegelmayer and R. Schrader, "Sentiment Polarity Classification using Statistical Data Compression Models," 2012 IEEE 12th International Conference on Data Mining Workshops, pp.731–738, 2012.

[20] P. Li, Q. Zhu, and W. Zhang, "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, pp.166–171, 2011.

[21] C. Quan, X. Wei, and F. Ren, "Combine Sentiment Lexicon and Dependency Parsing for Sentiment Classification," Proc. 2013 IEEE/SICE International Symposium on System Integration, pp.100–104, Kobe International Conference Center, Kobe, Japan, Dec. 15–17, 2013.

[22] 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.

[23] 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.

[24] K. Dave, S. Lawrence, and D.M. Pennock, "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews," Proc. WWW, pp.519–528, 2003.

[25] T.S. Bang, C. Haruechaiyasak, and V. Sornlertlamvanich, "Vietnamese Sentiment Analysis based on Term Feature Selection Approach," Proc. 10th International Conference on Knowledge Information and Creativity Support Systems (KICSS 2015), 12-14 Nov. 2015, pp.196–204, 2015.

[26] N.T. Duyen, N.X. Bach, and T.M. Phuong, "An Empirical Study on Sentiment Analysis for Vietnamese," 2014 International Conference on Advanced Technologies for Communications, pp.309–314, 2014.

[27] D.Q. Nguyen, D.Q. Nguyen, S.B. Pham, P.-T. Nguyen, and M.L. Nguyen, "From Treebank Conversion to Automatic Dependency Parsing for Vietnamese," Natural Language Processing and Information Systems, Lecture Notes in Computer Science, vol.8455, pp.196–207, Springer International Publishing, Cham, 2014.

[28] D.J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2005.

[29] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Constructing Free-Energy Approximations and Generalized Belief Approximation Algorithms," IEEE Trans. Inf. Theory, vol.51, no.7, pp.2282–2312, 2005.

[30] T. Laurence C., A Vietnamese Reference Grammar, University of Hawaii Press, 1987. ISBN 9780824811174.

[31] A.-C. Le, P.-T. Nguyen, H.-T. Vuong, M.-T. Pham, and T.-B. Ho, "An Experimental Study on Lexicalized Statistical Parsing for Vietnamese," 2009 International Conference on Knowledge and Systems Engineering, pp.162–167, 2009.

[32] P.-T. Nguyen, X.-L. Vu, T.-M.-H. Nguyen, V.-H. Nguyen, and H.-P. Le, "Building a Large Syntactically-Annotated Corpus of Vietnamese," Proc. 3rd Linguistic Annotation Workshop (LAW) at ACL-IJCNLP '09, pp.182–185, 2009.

[33] T.H. Duong, M.Q. Tran, and T.P.T. Nguyen, "Collaborative Vietnamese WordNet Building using Consensus Quality," Vietnam Journal of Computer Science, vol.4, no.2, pp.85–96, 2017.

Tran Sy Bang received a B.S. degree in Information and Communication Technology from the Asian Institute of Technology in 2014, Thailand. Currently, he is pursuing an M.S. degree in Information and Communication Technology from the Thailand Advanced Institute of Science and Technology and Tokyo Institute of Technology (TAIST-Tokyo Tech), Sirindhorn International Institute of Technology, Thammasat University, Thailand.

Virach Sornlertlamvanich received his D.Eng. in Computer Science at the Tokyo Institute of Technology, Japan. He received the National Distinguished Researcher awarded in 2003, in Information Technology and Communication from The National Research Council of Thailand. His research interests are Natural Language Processing, Artificial Intelligence, Datamining, Social Media Understanding, Asian WordNet, Digitized Thailand, and Language Resources.