A Graph-based Approach for Sentiment Sentence Extraction

Kazutaka Shimada, Daigo Hashimoto and Tsutomu Endo

Department of Artificial Intelligence, Kyushu Institute of Technology
680-4 Iizuka Fukuoka 820-8502 Japan
{shimada, d_hashimoto, endo}@pluto.ai.kyutech.ac.jp

Abstract. As the World Wide Web rapidly grows, a huge number of online documents are easily accessible on the Web. We obtain a huge number of review documents that include user's opinions for products. To classify the opinions is one of the hottest topics in natural language processing. In general, we need a large amount of training data for the classification process. However, construction of training data by hand is costly. In this paper, we examine a method of sentiment sentence extraction. This task is to classify sentences in documents into opinions and non-opinions. For the task, we use the Hierarchical Directed Acyclic Graph (HDAG) proposed by Suzuki et al. We obtained high accuracy in the sentiment sentence extraction task. The experimental result shows the effectiveness of the method based on the HDAG.

Keywords : Sentiment Analysis, Sentiment Sentence Extraction, Graph-based Approach, Hierarchical Directed Acyclic Graph, Similarity

1 Introduction

As the World Wide Web rapidly grows, a huge number of online documents are easily accessible on the Web. Finding information relevant to user needs has become increasingly important. The most important information on the Web is usually contained in the text. We obtain a huge number of review documents that include user's opinions for products. Buying products, users usually survey the product reviews. More precise and effective methods for evaluating the products are useful for users. To classify the opinions is one of the hottest topics in natural language processing. Many researchers have recently studied extraction and classification of opinions [8, 11, 12, 16, 17].

There are many research areas for sentiment analysis; extraction of sentiment expressions, identification of sentiment polarity of sentences, classification of review documents and so on. In this paper, we focus on sentiment sentence extraction. Extraction of sentiment expressions or sentiment sentences is one of the most important tasks in the sentiment analysis because classification tasks usually need a large amount of training data to generate a high accuracy classifier. There are several reports for classification of sentences [9, 11]. However, the purpose of these studies is to classify sentences into positive and negative opinions. Our purpose in this paper is to classify sentences into opinions and non-opinions. Touge et al. [15] and Kawaguchi et al. [7] have proposed methods for opinion extraction. However, these approaches essentially need a large amount of training data for the process. Construction of training data by hand is costly. Kaji and Kitsuregara have reported a method of acquisition of sentiment sentences in HTML documents [5]. The method required only several rules by hand and obtained high accuracy. Also they have proposed a method for building lexicon for sentiment analysis [6]. The knowledge extracted from the Web by using the proposed methods contains the huge quantities of words and sentences. Takamura et al. also have reported a method for extracting polarity of words [14]. These dictionaries are versatile and valuable for users because they do not depend on a specific domain. Here, assume that we need to construct a system for a domain. In that case, we often desire domain-specific knowledge for the system. Therefore, we need to efficiently extract sentiment sentences, which depend on a particular domain or topic.

In this paper, we propose a method of sentiment sentence extraction. The method can deal with domain specific and nonspecific areas. Also our method does not require dictionaries of sentiment expressions. It uses several sample sentences for the extraction process. In the process, we compute a similarity between the sample sentences and target sentences. For the similarity calculation, we employ the graph-based approach, called Hierarchical Directed Acyclic Graph (HDAG), which has been proposed by Suzuki et al [13].

In Section 2, we explain the HDAG data structure and layers. In Section 3, we describe a sentiment sentence extraction process with similarity calculation based on the HDAG. In Section 4, we evaluate the performance of the method and conclude this paper in Section 5.

2 A Graph-based Data Structure

In this section, we explain a graph-based data structure to compute a similarity.

2.1 Hierarchical Directed Acyclic Graph

In natural language processing, bag-of-words representation is the most general way to express features of a sentence for the similarity calculation. However, it is insufficient to represent the features of a sentence because of lack of relations between words. To solve this problem, many researchers have proposed new approaches: a string kernel [10], a word-sequence kernel [1] , an extended string subsequence kernel [3] and a tree kernel [2] . These kernels are usually more effective as compared with bag-of-words based methods. However, they are not the best representation for deep and complex features, such as semantic or grammatical information, in a sentence because they are somewhat of a simple representation.

To solve the problems, Suzuki et al. have reported a new graph-based approach, called Hierarchical Directed Acyclic Graph kernels (HDAG) [13]. The method can handle many linguistic features in a sentence and includes characteristics of tree and sequence kernels. The HDAG is a hierarchized graph-in-graph structure. It represents semantic or grammatical information in a sentence. In this paper, we use the HDAG structure for the sentiment sentence extraction. We compute a similarity between HDAGs generated from sentences. See [13] for more information about the HDAG.

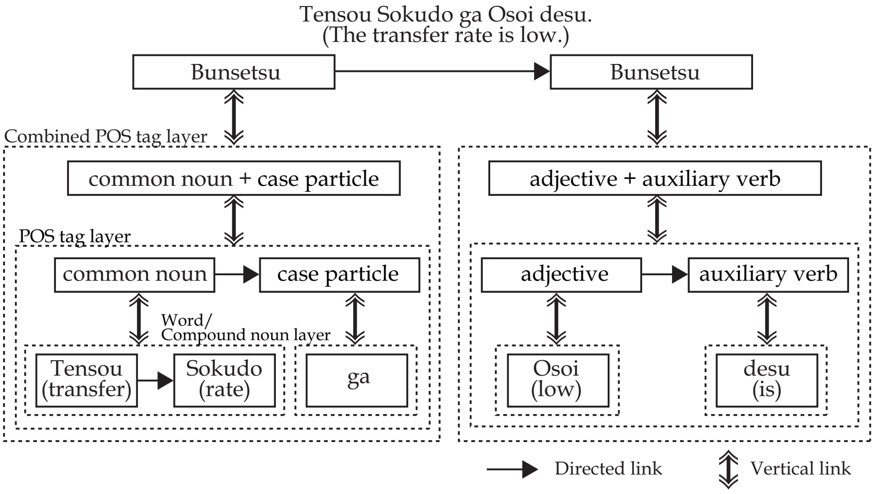

Fig. 1. An example of an HDAG expression.

2.2 Layer

Layers in the HDAG denote semantic or grammatical information in a sentence. To compute similarity between sentences correctly, we add new layers to the original and naive HDAG. The HDAG in this paper consists of three layers as follows:

- **Combined POS tag layer**
  This layer consists of part of speech tags of words. We unify the POS tags of words in a bunsetsu[1] into one node. Roughly speaking, this layer expresses sort of semantic information about each bunsetsu.
- **POS tag layer**
  This layer consists of the POS tags of each word or each compound noun.
- **Word/Compound noun layer**
  This layer contains two roles; the layer for words and compound nouns. The layer for each word contains the surface expression of a word. We can use the surface information for calculation of similarity by adding this layer. The 2nd role is handling compound nouns in bunsetsus. This layer often resolves a problem of difference between surface expressions. We unify nouns belonging to a compound noun and then dispose it under the POS node of its compound nouns. For example, we flexibly treat the difference of the following expressions in similarity calculation by adding this layer: "file downloading software", "downloading software" and "software".

Figure 1 shows an example of an HDAG expression in this paper. In the HDAG, the elements, such as "Bunsetsu" and "Common noun", in each rectangle are the attributes of each node. The directed links are a kind of the dependency relation between elements. The double-headed arrows denote the link between a node and a sub-graph enclosed with a dashed line.

[1] A bunsetsu is a linguistic unit in Japanese. It usually consists of one content word and its function words.

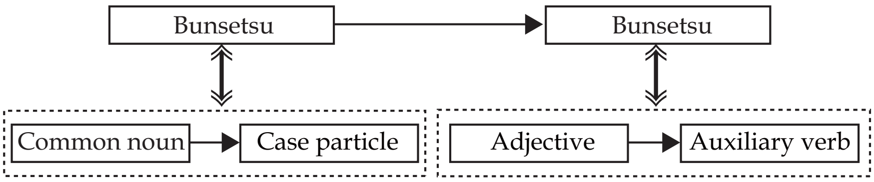

Fig. 2. An example of a graph structure.

3 Similarity Calculation

In this section, we explain a method of similarity calculation based on the HDAG structure. First, we describe a conversion method of sentences into the HDAGs. Next, we explain an extraction method of hierarchical attribute subsequences from HDAG structures for the similarity calculation. Finally, we introduce a method of similarity calculation and the extraction process using it.

3.1 Preprocessing

There are two processes as the preprocessing for similarity calculation; conversion and extraction of hierarchical attribute subsequences. First we explain the conversion process. To convert sentences into the HDAG structure, we need to analyze them, that is morphological analysis and dependency analysis. In this paper we use JUMAN[2] as the morphological analyzer and KNP[3] as the dependency analyzer. Figure 2 shows the graph structure[4] generated from the sentence "Onshitu ga Ii Desu. (The sound quality is good.)[5]."

Next we need to extract hierarchical attribute subsequences for the similarity calculation. A hierarchical attribute subsequence is an attribute list with hierarchical structures. The similarity is computed from corresponding hierarchical attribute subsequences extracted from sentences that we want to compare.

Here Suzuki et al. [13] introduced two factors; β and λ. The β ($\beta > 0$) is the factor for the correspondence. The value of each hierarchical attribute sequence is multiplied by $\sqrt{\beta}^m$ where m represents the number of attributes in the hierarchical attribute sequence. The λ is the decay factor λ ($0 \leq \lambda \leq 1$). The system allows not only exactly matching structures but also similar structures by using this factor. The actual decay value of a skipping node v_i is $\Lambda(v_i) = \lambda^{n+1}$ where n is the number of nodes in a graph G if vertical link exists, or $\Lambda(v) = \lambda$ otherwise.

Figure 3 shows an example of hierarchical attribute subsequences and the factors. In the figure, a dependency relation and a hierarchical relation are expressed by using a comma and a nested structure, respectively. For the two HDAGs, the hierarchical attribute subsequence <Bunsetsu<Noun,Particle>, Bunsetsu<Adj,Aux>> appears in both

[2] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html
[3] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html
[4] In this simplified explanation, the graph structure for an example is expressed without the layers described in the previous section.
[5] Onshitu is a noun (sound quality), ga is a case particle, Ii is an adjective (good), and Desu is an auxiliary verb (is).

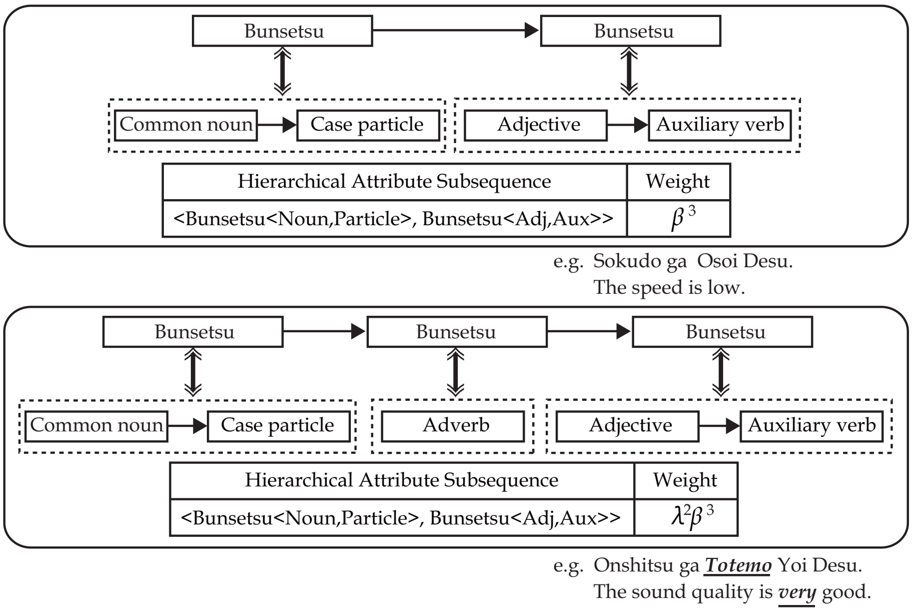

Fig. 3. An example of extraction of corresponding hierarchical attribute subsequences and the weight.

sentences. Since the number of attributes in the hierarchical attribute subsequence is 6, the value of β is $\sqrt{\beta}^6 = \beta^3$. The hierarchical attribute subsequence of the 2nd sentence in the figure is generated by skipping a node, Adverb. Therefore, the weight contains $\lambda^{1+1} = \lambda^2$. These weights are used in the similarity calculation process.

3.2 Similarity Between Two Sentences

Next, we compute a similarity between two HDAG structures. First, we search the common hierarchical attribute subsequences between HDAGs (See Figure 3). Then we multiply the weight values of them. For example, the correspondence of them in Figure 3 is $\lambda^2\beta^6$. Finally, we divide the sum total of correspondence values by the product of the numbers of bunsetsus of the two sentences. We handle this value as the similarity between them.

Here we consider the factor β. Suzuki et al. [13] defined the range of the β as $0 < \beta \leq 1$. However, we define the range as $\beta > 0$ in this paper. Also we categorize the range into two types; $0 < \beta \leq 1$ and $\beta > 1$. Our method computes the similarity focusing on structural information if $\beta \leq 1$. If the β is more than 1, it computes the similarity focusing on surface expressions. This is due to layers that we constructed. In our layers, the word layer and compound noun layer are lower layer than the structural layer, i.e., the POS tag layer. Therefore surface expressions are treated as important element in the case that $\beta > 1$ because the elements in deeper layers possess high weight values. We apply these two types of the parameter β into our method.

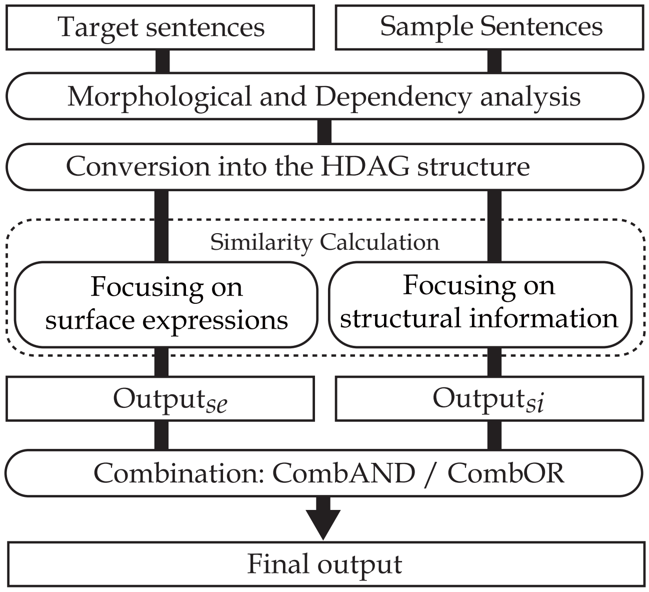

Fig. 4. The outline of the sentence extraction process.

3.3 Sentence Extraction

In this subsection, we explain the sentence extraction process based on the HDAG and the similarity calculation. The process is as follows:

1. prepare sample sentences as seeds for similarity calculation,
2. compute the similarity between each seed and target sentences,
3. extract n-best lists of each seed as sentiment sentence lists,
4. combine n-best lists obtained by two different parameters of β.

For the combination in the last step, we compare two strategies.

CombAND We extract the intersection of each n-best list as the output.
CombOR We extract the union of each n-best list as the output.

Figure 4 shows the outline of the extraction process.

4 Experiment

In this section we evaluated the proposed method with a review document set.

4.1 Dataset and Criteria

We used review documents of a portable audio player[6] posted in the bulletin board system of kakaku.com[7]. We extracted 1052 Japanese sentences from the review documents.

[6] SONY Walkman NW-A808
[7] http://www.kakaku.com/

Table 1. The experimental result

	$Sent_{real}$	$Sent_{non}$	Acc
Structural information	32	4	0.889
Surface expressions	43	4	0.915
CombAND	22	**1**	**0.957**
CombOR	**53**	7	0.883
BOW (Baseline)	42	7	0.857

The dataset consists of 610 sentiment sentences and 442 non-sentiment sentences. For the experiment, we prepared 10 sample sentences as seeds for the sentence extraction process. All the seed sentences in this experiment were sentiment sentences. We generated the seed sentences on the basis of some evaluation criteria which were mentioned in the review documents; e.g., "design of the product", "Sound quality" and so on.

In this experiment, we set $\lambda = 0.5$. Also we set $\beta = 0.5$ as the parameter for focusing on structural information and $\beta = 1.5$ as the parameter for focusing on surface expressions. The number of sentences we extracted in this experiment is 5 for each seed sentence, that is 5-best list. In other words, we extracted the top 5 sentences that possessed high similarity as the sentiment sentences that were estimated from each sample sentences. We did not employ any thresholds for the similarity in the extraction process.

We used the following three criteria for this evaluation.

- $Sent_{real}$: This criterion is the number of sentiment sentences extracted correctly from target sentences.
- $Sent_{non}$: This criterion is the number of non-sentiment sentences extracted from target sentences.
- Acc: This criterion is the accuracy computed from $Sent_{real}$ and $Sent_{non}$.

$$Acc = \frac{Sent_{real}}{Sent_{real} + Sent_{non}}$$

Note that we omitted same sentences in the output from the proposed method when we counted $Sent_{real}$ and $Sent_{non}$ in this experiment.

4.2 Results

Table 1 shows the experimental result. In the table, the BOW denotes a similarity calculation method based on the COS measure and bag-of-words features. This is a baseline in this experiment. The accuracy rates of each approach in our method outperformed the baseline method based on BOW features. Our methods obtained high accuracies even without combinations, namely **CombAND** and **CombOR**. In addition, the method focusing on surface expressions (SE) outperformed the method focusing on structural information (SI) in terms of all criteria. Table 2 shows the top 3 sentences extracted from target sentences in the case that the seed sentence was "The sound quality is good.".

For the combinations, the accuracy of the **CombOR** was the lowest of the methods although the number of sentiment sentences extracted correctly was the best of

Table 2. The extracted sentences (translated into English)

Rank	SI ($\beta \leq 1$)	SE ($\beta > 1$)
1	The sound quality is barely good.	The sound quality is barely good.
2	The display is easily viewable.	The sound quality is wonderful.
3	The machine body is somewhat heavy.	The sound quality is great.

them. On the other hand, the accuracy of the **CombAND** produced the best performance. Although the number of extracted sentences with the **CombAND** drastically decreased, the output possessed high reliability.

Besides, our method usually obtained long sentences as compared with seed sentences. The average lengths of seed sentences and output sentences were 5.5 and 9.1 words respectively. The maximum length in the output sentences was 27 words. This result shows that our method can extract great variety of sentiment sentences. The following sentences are the instances of seed sentences and their output sentences:

Seed$_1$ I am almost satisfied with this product (Zentai-teki ni manzoku dekiru seihin desu).

Output$_1$ I think that a good thing equipped with the function that I want was released (Yatto watashi ga nozomu kinou ga zyuzitusita yoi mono ga deta to iu kanzi desu).

Seed$_2$ It is difficult to push the play button (Saisei botan ga oshi nikui desu).

Output$_2$ It is not convenient for WinMediaPlayer users to use attached music file transfer software XYZ (Sen-you no ongaku fairu tensou sohuto XYZ ha WinMediaPlayer kara no norikae niha tukai durai desu).

5 Discussion and Conclusions

In this paper, we proposed a method of sentiment sentence extraction based on a graph-based approach, called Hierarchical Directed Acyclic Graph. Our method can extract sentiment sentences with several sample sentences. We obtained high accuracy in the experiment. However, the number of extracted sentences was not enough, that is the recall rate was low (less than 10%).

One of the solutions of this problem is to apply a bootstrapping approach into our method. We might acquire more sentiment sentences by adding the extracted sentences as new seeds for the extraction process because the accuracy of CombAND was extremely high. To use CombOR is one of the ideas to extract large quantities of sentences in the case that the number of sentences extracted in the bootstrapping process is saturated, i.e., the final step of the bootstrapping approach. Another approach to improve the recall rate is use of the extracted sentences for the training data of the sentiment classification task. Wiebe and Riloff [18] have proposed a method for creating subjective and objective classifiers from unannotated texts. They used some rules for constructing initial training data. Then they used the data for generating a classifier. We think that the outputs from our method also can be used for the training data of a classifier for this sentiment sentence classification task. The value of n of the n-best list is the important factor for the improvement of the recall rate although the accuracy decreases. For the **CombOR** , the accuracy and the $Sent_{real}$ were 0.803 and 94 in the case that the n was 10. Also the accuracy and the $Sent_{real}$ were 0.773 and 126 in the case that the n was 15. We need to discuss the appropriate number of sentences that we extract in our method. Our method depends on seed sentences. If the seed sentences are changed, the accuracy also changes. We examined other seed sentences after the experiment in the previous section. As a result, the accuracy fluctuated; approximately $\pm 5\%$. To generate appropriate seed sentences is one of the most important tasks for our method.

In the previous section, we evaluated our method with fixed parameters. However, these are not always the best parameter values. The parameters β and λ are important factors for the similarity calculation. We compared several values of these parameters: $\lambda = \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$, $\beta = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ for the SI and $\beta = \{1.1, 1.3, 1.5, 1.7, 1.9\}$ for the SE. The best accuracy rates of the SI and the SE in this dataset were 0.950 ($\lambda = 0.9, \beta = 0.9$) and 0.975 ($\lambda = 1.0, \beta = 1.1$) respectively. However, these values depend on the dataset in the experiment. Although we evaluated these parameters with another dataset, the method with the parameters did not produce the best accuracy. Therefore we need to consider the automatic determination of these parameters. The average accuracy rates of each parameter in the dataset, which were used in the previous section, were 0.867 for the SI and 0.918 for the SE respectively. The standard deviation values were 0.048 for the SI and 0.025 for the SE. These results show that our method provides high and stable accuracy.

Our future work includes (1) evaluation of our method in a large-scale dataset and other datasets, (2) improvement of the accuracy by adding other layers to the HDAG structure, such as semantic features of words [4], and (3) construction of a sentiment sentence maintenance tool based on this approach.

References

[1] N. Cancedda, E. Gaussier, C. Goutte, and J. M. Renders. Word-sequence kernels. *J. Machine Learning Research*, 3:1059–1082, 2003.

[2] M. Collins and N. Duffy. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14*, 2002.

[3] Tsutomu Hirao, Jun Suzuki, Hideki Isozaki, and Eisaku Maeda. Dependency-based sentence alignment for multiple document summarization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)*, 2004.

[4] S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi, editors. *Goi-Taikei . A Japanese Lexicon (in Japanese)*. Iwanami Shoten, 1997.

[5] Nobuhiro Kaji and Masaru Kitsuregawa. Automatic construction of polarity-tagged corpus from html documents. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING/ACL2006)*, pages 452–459, 2006.

[6] Nobuhiro Kaji and Masaru Kitsuregawa. Building lexicon for sentiment analysis from massive html documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-CoNLL2007)*, 2007.

[7] Toshihiro Kawaguchi, Tohgoroh Matsui, and Hatato Ohwada. Opinion extraction from weblog using svm and newspaper article (in Japanese). In *The 20th Annual Conference of the Japanese Society for Artificial Intelligence*, 2006.

[8] N. Kobayashi, R. Iida, K. Inui, and Y. Matsumoto. Opinion extraction using a learning-based anaphora resolution technique. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 175–180, 2005.

[9] Taku Kudo and Yuji Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.

[10] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C.Watkins. Text classification using string kernel. *J. Machine Learning Research*, 2:419–444, 2002.

[11] I. Osajima, K. Shimada, and T. Endo. Classification of evaluative sentences using sequential patterns. In *Proceedings of the 11nd Annual Meeting of The Association for Natural Language Processing (in Japanese)*, 2005.

[12] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.

[13] Jun Suzuki, Yutaka Sasaki, and Eisaku Maeda. Hierarchical directed acyclic graph kernel. *Systems and Computers in Japan*, 37(10):58–68, 2006.

[14] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL2005)*, pages 133–140, 2005.

[15] Yasunari Touge, Kazuteru Ohashi, and Kazuhide Yamamoto. Extracting opinion sentence adapted to topic using iteration learning (in Japanese). In *IPSJ SIG Notes*, pages 43–50, 2004.

[16] K. Tsutsumi, K. Shimada, and T. Endo. Movie review classification based on a multiple classifier. In *the 21th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, 2007.

[17] P. D. Turney. Thumbs up? or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, 2002.

[18] Janyce Wiebe and Ellen Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005)*, 2005.