Sentiment Sentence Extraction Using a Hierarchical Directed Acyclic Graph Structure and a Bootstrap Approach [*]

Kazutaka Shimada[a], Daigo Hashimoto[a], and Tsutomu Endo[a]

[a] 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. The goal of our study is to construct a sentiment tagging tool for particular domains. In this paper, we propose a method of sentiment sentence extraction for the 1st step of the system. For the task, we use a Hierarchical Directed Acyclic Graph (HDAG) structure. We obtained high accuracy with the graph based approach. Furthermore, we apply a bootstrap approach to the sentiment sentence extraction process. The experimental result shows the effectiveness of the method.

Keywords: Sentiment analysis, sentence extraction, similarity, HDAG.

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 (Hatzivassiloglou and McKeown 1997, Kobayashi et al. 2005, Pang et al. 2002, Wiebe and Riloff 2005).

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. The goal of our study is to easily construct a corpus for sentiment information for particular domains that users want. For the purpose we need to extract sentiment sentences from documents as the 1st step of the corpus construction. 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 (Kudo and Matsumoto 2004, Osajima et al .2005). 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. (2004) and Kawaguchi et al. (2006) 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 (2006) have reported a method of acquisition of sentiment sentences in HTML documents. The method required only several rules by hand and obtained high accuracy. Also they have proposed a method for building lexicon for sentiment analysis (Kaji and Kitsuregawa 2007). The knowledge extracted from the Web by using the proposed methods contains the huge quantities of words and sentences. Takamura et al. (2005) also have reported a method for extracting polarity of words. 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. It uses several sample sentences for the extraction process. In the process, we compute a similarity between the sample sentences and target sentences. Yu and Hatzivassiloglou (2003) have reported a similarity based method using words, phrases and WordNet synsets for sentiment sentence extraction. However, word-level features are not always suitable for the extraction process because of lack of relations between words. For the similarity calculation we, therefore, employ the graph-based approach, called Hierarchical Directed Acyclic Graph (HDAG), which has been proposed by Suzuki et al. (2006). Furthermore, we apply a bootstrap approach into the sentiment sentence extraction process. The number of extracted sentiment sentences increases with the bootstrap approach.

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 the problems, Suzuki et al. (2006) have reported a new graph-based approach, called Hierarchical Directed Acyclic Graph kernels (HDAG). 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 (Suzuki et al. 2006) for more information about the HDAG.

2.2. Layer

Layers in the HDAG denote grammatical information in a sentence. To compute similarity between sentences correctly, we add some layers to a naive HDAG structure. 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. This layer expresses a pack of POS tags in 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 the surface expression of each word. We can use the surface information for calculation of similarity by adding this layer. The layer is used for handling compound

---

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

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.
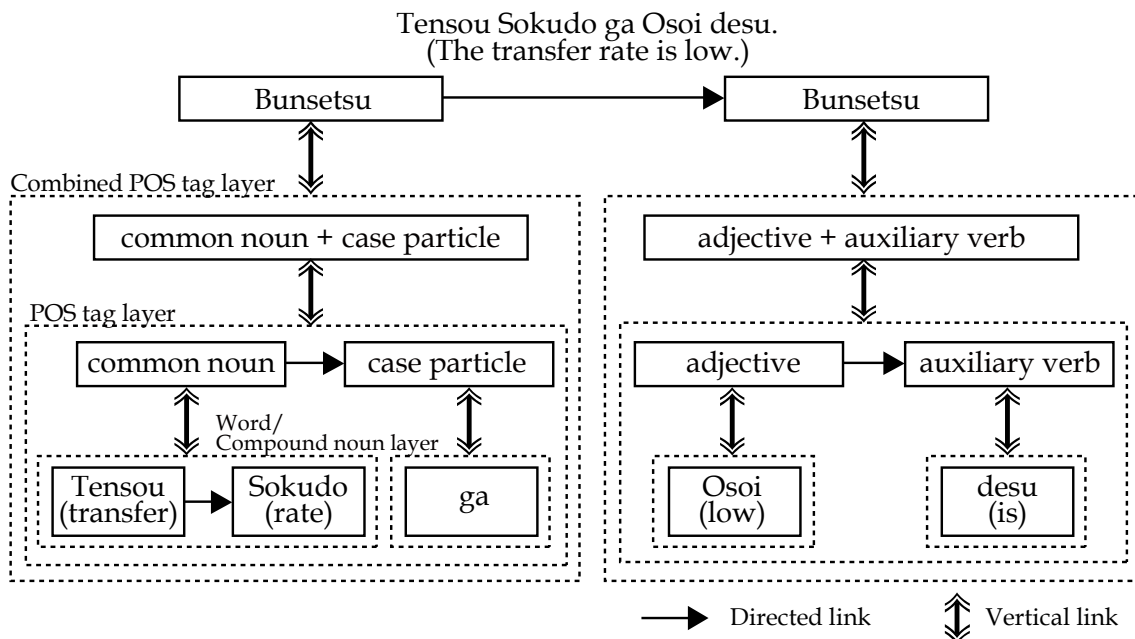

Figure 1: An example of an HDAG expression.

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.

  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 correspondence of hierarchical attribute subsequences extracted from sentences that we want to compare.

---

[2] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html

[3] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html

Here Suzuki et al. (2006) introduced two factors; β and γ. The β ($0 < \beta \le 1$) 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 ($0 \le \lambda \le 1$). 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 if vertical link exists, or $\Lambda(v) = \lambda$ otherwise.

Figure 2 shows an example of hierarchical attribute subsequences and the factors[4]. 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 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.
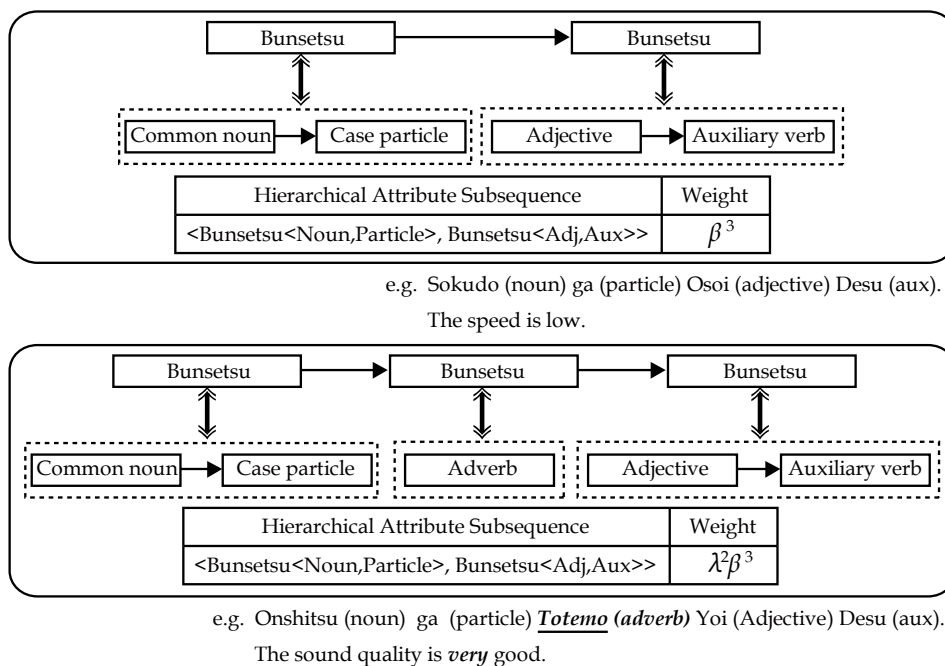

e.g. Sokudo (noun) ga (particle) Osoi (adjective) Desu (aux).

The speed is low.

e.g. Onshitsu (noun) ga (particle) *Totemo (adverb)* Yoi (Adjective) Desu (aux).

The sound quality is *very* good.

Figure 2: An example of corresponding hierarchical attribute subsequences and the weight.

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 2). Then we multiply the weight values of them. For example, the correspondence of them in Figure 2 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.

---

[4] In this simplified explanation, the graph structure for an example is expressed without the layers described in the previous section.

Here we consider the factor β. Suzuki et al. (2006) defined the range as $0 < \beta \leq 1$ to avoid overfitting in machine learning methods. 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 (SI) if $\beta \leq 1$. If the β is more than 1, it computes the similarity focusing on surface expressions (SE). 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 to our method.

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.
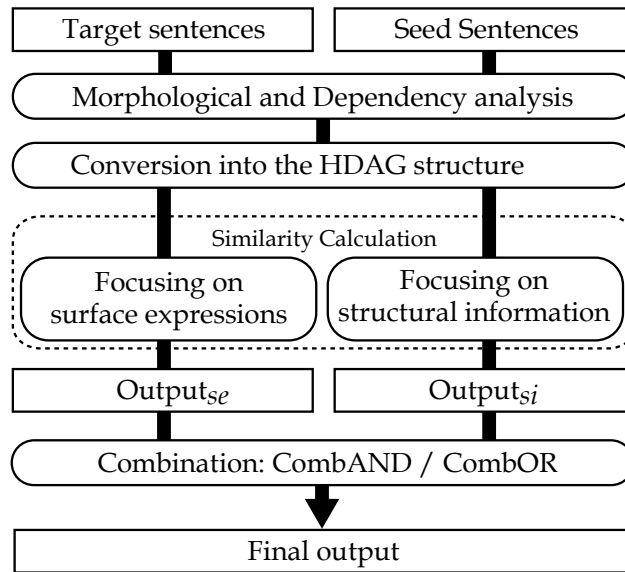
Figure 3 shows the outline of the extraction process.

Figure 3: The outline of the sentence extraction process.

4. Bootstrap Approach

The method proposed in the previous section contains a problem; the number of sentences extracted from target sentences. It depends on the number of seed sentences and the number of n of n-best. Preparing many seed sentences is usually high cost for users. If the number of n becomes large, the number of extracted sentences increases. However, it leads to decrease of the accuracy.

To solve this problem, we apply a bootstrap approach to our method. In the bootstrap approach, we use both CombAND and CombOR as the situation demands. We use the CombAND to acquire new seed sentences because it usually generates the high accuracy. On the other hand,

we use the CombOR in the final step of the bootstrap process because it usually extracts more sentences than the CombAND. In addition, we change the value of n in each step of the bootstrap approach. In other words, we decrease the value in the bootstrap process. In the bootstrap approach, the method often extracts incorrect sentences as sentiment sentences, and add them into the seed sentence list. The method tends to extract incorrect sentences in the later steps of the bootstrap process. Therefore we set a larger value to the n in the early steps and a smaller value to it in the later steps.

The process is as follows:
1. extract sentences with CombAND by using current seed sentences,
2. add the extracted sentences into the seed sentence list as new seeds,
3. decrease the value of n,
4. iterate the process 1-3, m-times,
5. extract sentences with CombOR as the final output.
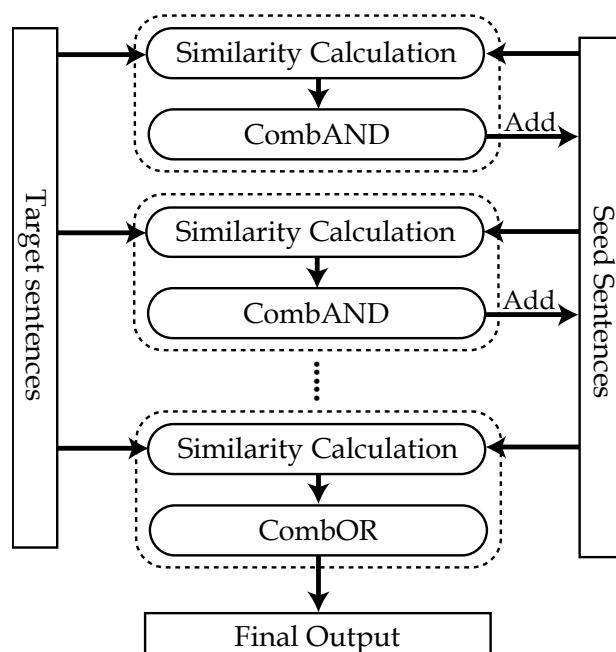
Figure 4 shows the outline of the bootstrap process.

Figure 4: The outline of the bootstrap approach process.

5. Experiment

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

5.1. Dataset and Criteria

We used review documents of a portable audio player[5] posted in the bulletin board system of kakaku.com[6]. We extracted 1052 Japanese sentences from the review documents. 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.

---

[5] SONY Walkman NW-A808

[6] http://www.kakaku.com/

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. For the bootstrap approach, we set the number of iterations as 3. The values of n in each step were $n=7$ in the 1st step, $n=5$ in the 2nd step and $n=3$ in the final step, respectively.

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 $Sent_{non}$.

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

- $Recall$: This criterion is the recall rate computed from $Sent_{real}$ and the number of sentiment sentences in the dataset.

5.2. Experimental Results

Table 1 shows the experimental result. In the table, the BOW denotes a similarity calculation method based on the cosine measure and bag-of-words features. This is a baseline in this experiment. The accuracy rates of each non-bootstrap 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.

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 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.

By using the bootstrap approach, the number of correct sentences nearly tripled although the accuracy decreased somewhat. This result shows the effectiveness of applying the bootstrap approach to our method. As an additional experiment, we evaluated our method with a fixed n; $n=5$, that is an experiment without decrease of n in each step. In the additional experiment, the accuracy rate became 0.78. This result shows the importance of changing n in each step of the bootstrap approach. Therefore we need to consider the determination of the appropriate n in each step.

One reason of the decrease of the accuracy with the bootstrap approach is that our method occasionally added non-sentiment sentences as seed sentences in each step. One of the solutions to this problem is to apply feedback from users to the bootstrap process, such as relevance feedback. This is one future work for our method.

The number of extracted sentences increased by using the bootstrap approach. However, the recall rate was insufficient. It was 0.256 even if we applied the bootstrap approach to the extraction process. One approach to improve the recall rate is use of the extracted sentences for the training data of the sentiment classification task. Wiebe and Riloff (2005) 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.

Table 1: The experimental result.

	$Sent_{real}$	$Sent_{non}$	Acc	Recall
Structural information	32	4	0.889	0.052
Surface expression	43	4	0.915	0.070
CombAND	22	*1*	*0.957*	0.002
CombOR	53	7	0.883	0.087
Bootstrap	*156*	30	0.839	*0.256*
BOW (Baseline)	42	7	0.857	0.069

6. Conclusions

In this paper, we proposed a method of sentiment sentence extraction based on a graph-based approach, called Hierarchical Directed Acyclic Graph (HDAG). Our method can extract sentiment sentences by using 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 extremely low. To solve this problem, we applied a bootstrap approach to out method. As a result, we acquired more sentiment sentences by adding the extracted sentences as new seeds. By using the bootstrap approach based on the combination of CombAND and CombOR, we obtained large quantities of sentences as compared with the non-bootstrap approach. However, even the recall rate with the bootstrap approach was insufficient. Therefore we need to discuss solutions to the problem. One approach is to apply feedback from users to the method. We need to consider a semi-automatic approach based on human-aid to accomplish high recall and accuracy rates. Some researchers have reported approaches with structured features (Zhang et al. 2006 and Zhou et al. 2007). We need to compare our method with them.

In this experiment, we evaluated our method with fixed parameters. However, they are not always the best parameter values. In previous work, we compared several values of these parameters (Shimada et al. 2008). By tuning the parameters, we obtained higher accuracy for the SI and SE. However, these values depended on the dataset in the experiment. Therefore we need to consider an automatic determination method of these parameters to achieve higher accuracy for the bootstrap approach.

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 (Ikehara et al. 1997), and (3) construction of a sentiment sentence maintenance tool based on this approach.

References

Hatzivassiloglou, V. and K. R. McKeown. 1997. Predicting the Semantic Orientation of Adjectives. *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (ACL) and *the 8th Conference of the European Chapter of the Association for Computational Linguistics* (EACL), pp.174-181.

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

Kaji, N. and M. Kitsuregawa. 2006. Automatic Construction of Polarity-tagged Corpus from HTML Documents. *In Proceedings of the 21st International Conference on Computational Linguistics* (COLING/ACL2006), pages 452.459.

Kaji, N. and M. Kitsuregawa. 2007. Building Lexicon for Sentiment Analysis from Massive HTML Documents. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP-CoNLL2007).

Kawaguchi, T., T. Matsui and H. Ohwada. 2006. Opinion Extraction from Weblog Using SVM and Newspaper Article (in Japanese). *In The 20th Annual Conference of the Japanese Society for Artificial Intelligence*.

Kobayashi, N., R. Iida, K. Inui and Y. Matsumoto. 2005. Opinion Extraction Using a Learning-

based Anaphora Resolution Technique. *In Proceedings of the Second International Joint Conference on Natural Language Processing* (IJCNLP-05), pp. 175-180.

Kudo, T. and Y. Matsumoto. 2004. A Boosting Algorithm for Classification of Semi-structured Text. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP).

Osajima, I., K. Shimada and T. Endo. 2005. Classification of Evaluative Sentences Using Sequential Patterns. *In Proceedings of the 11nd Annual Meeting of The Association for Natural Language Processing* (in Japanese).

Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP), pp. 79-86.

Shimada, K., D. Hashimoto and T. Endo. 2008. A Graph-based Approach for Sentiment Sentence Extraction. *First International Workshop on Algorithms for Large-Scale Information Processing in Knowledge Discovery* (ALSIP 2008), Working Note, pp. 42-51.

Suzuki, J., Yutaka Sasaki and Eisaku Maeda. 2006. Hierarchical Directed Acyclic Graph Kernel. *Systems and Computers in Japan*, 37(10), pp. 58-68.

Takamura, H., T. Inui and M. Okumura.2005. Extracting Semantic Orientations of Words Using Spin Model. *In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (ACL2005), pp. 133-140.

Touge, Y., K. Ohashi and K. Yamamoto.2004. Extracting Opinion Sentence Adapted to Topic Using Iteration Learning (in Japanese). *In IPSJ SIG Notes*, pp. 43-50.

Turney, P. D. 2002. 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*, pp. 417-424.

Wiebe, J. and E. Riloff. 2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. *In Sixth International Conference on Intelligent Text Processing and Computational Linguistics* (CICLing-2005).

Yu, H. and V. Hatzivassiloglou. 2003. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP), pp.129-136.

Zhang, M., J. Zhang, J. Su, G. Zhou. 2006. A Composite Kernel to Extract Relations between Entities with Both Flat and Structured Features. *In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 825-832.

Zhou, G., M. Zhang, D. Ji and Q. Zhu. 2007. Tree Kernel-Based Relation Extraction with Context-Sensitive Structured Parse Tree Information. *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (EMNLP-CoNLL 2007), pp. 728-736.