

# Latent Variable Models for Semantic Orientations of Phrases

**Hiroya Takamura**

Precision and Intelligence Laboratory  
Tokyo Institute of Technology  
takamura@pi.titech.ac.jp

**Takashi Inui**

Japan Society of the Promotion of Science  
tinui@lr.pi.titech.ac.jp

**Manabu Okumura**

Precision and Intelligence Laboratory  
Tokyo Institute of Technology  
oku@pi.titech.ac.jp

## Abstract

We propose models for semantic orientations of phrases as well as classification methods based on the models. Although each phrase consists of multiple words, the semantic orientation of the phrase is not a mere sum of the orientations of the component words. Some words can invert the orientation. In order to capture the property of such phrases, we introduce latent variables into the models. Through experiments, we show that the proposed latent variable models work well in the classification of semantic orientations of phrases and achieved nearly 82% classification accuracy.

## 1 Introduction

Technology for affect analysis of texts has recently gained attention in both academic and industrial areas. It can be applied to, for example, a survey of new products or a questionnaire analysis. Automatic sentiment analysis enables a fast and comprehensive investigation.

The most fundamental step for sentiment analysis is to acquire the semantic orientations of words: desirable or undesirable (positive or negative). For example, the word “beautiful” is positive, while the word “dirty” is negative. Many researchers have developed several methods for this purpose and obtained good results (Hatzivassiloglou and McKeown, 1997; Turney and Littman, 2003; Kamps et al., 2004; Takamura et al., 2005; Kobayashi et al., 2001). One of the next problems to be solved is to acquire semantic orientations of phrases, or multi-term expressions. No computational model for semanti-

cally oriented phrases has been proposed so far although some researchers have used techniques developed for single words. The purpose of this paper is to propose computational models for phrases with semantic orientations as well as classification methods based on the models. Indeed the semantic orientations of phrases depend on context just as the semantic orientations of words do, but we would like to obtain the most basic orientations of phrases. We believe that we can use the obtained basic orientations of phrases for affect analysis of higher linguistic units such as sentences and documents.

The semantic orientation of a phrase is not a mere sum of its component words. Semantic orientations can emerge out of combinations of non-oriented words. For example, “light laptop-computer” is positively oriented although neither “light” nor “laptop-computer” has a positive orientation. Besides, some words can invert the orientation of a neighboring word, such as “low” in “low risk”, where the negative orientation of “risk” is inverted to a “positive” by the adjective “low”. This kind of *non-compositional* operation has to be incorporated into the model. We focus on “noun+adjective” in this paper, since this type of phrase contains most of interesting properties of phrases, such as emergence or inversion of semantic orientations.

In order to capture the properties of semantic orientations of phrases, we introduce latent variables into the models, where one random variable corresponds to nouns and another random variable corresponds to adjectives. The words that are similar in terms of semantic orientations, such as “risk” and “mortality” (i.e., the positive orientation emerges when they are “low”), make a cluster in these models. Our method is language-

independent in the sense that it uses only cooccurrence data of words and semantic orientations.

## 2 Related Work

We briefly explain related work from two viewpoints: the classification of word pairs and the identification of semantic orientation.

### 2.1 Classification of Word Pairs

Torisawa (2001) used a probabilistic model to identify the appropriate case for a pair of words constituting a noun and a verb with the case of the noun-verb pair unknown. Their model is the same as Probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 2001), which is a generative probability model of two random variables. Torisawa's method is similar to ours in that a latent variable model is used for word pairs. However, Torisawa's objective is different from ours. In addition, we used not the original PLSI, but its expanded version, which is more suitable for this task of semantic orientation classification of phrases.

Fujita et al. (2004) addressed the task of the detection of incorrect case assignment in automatically paraphrased sentences. They reduced the task to a problem of classifying pairs of a verb and a noun with a case into correct or incorrect. They first obtained a latent semantic space with PLSI and adopted the nearest-neighbors method, in which they used latent variables as features. Fujita et al.'s method is different from ours, and also from Torisawa's, in that a probabilistic model is used for feature extraction.

### 2.2 Identification of Semantic Orientations

The semantic orientation classification of words has been pursued by several researchers (Hatzivassiloglou and McKeown, 1997; Turney and Littman, 2003; Kamps et al., 2004; Takamura et al., 2005). However, no computational model for semantically oriented phrases has been proposed to date although research for a similar purpose has been proposed.

Some researchers used sequences of words as features in document classification according to semantic orientation. Pang et al. (2002) used bigrams. Matsumoto et al. (2005) used sequential patterns and tree patterns. Although such patterns were proved to be effective in document classification, the semantic orientations of the patterns

themselves are not considered.

Suzuki et al. (2006) used the Expectation-Maximization algorithm and the naive bayes classifier to incorporate the unlabeled data in the classification of 3-term evaluative expressions. They focused on the utilization of context information such as neighboring words and emoticons. Turney (2002) applied an internet-based technique to the semantic orientation classification of phrases, which had originally been developed for word sentiment classification. In their method, the number of hits returned by a search-engine, with a query consisting of a phrase and a seed word (e.g., "*phrase* NEAR *good*") is used to determine the orientation. Baron and Hirst (2004) extracted collocations with Xtract (Smadja, 1993) and classified the collocations using the orientations of the words in the neighboring sentences. Their method is similar to Turney's in the sense that cooccurrence with seed words is used. The three methods above are based on context information. In contrast, our method exploits the internal structure of the semantic orientations of phrases.

Inui (2004) introduced an attribute *plus/minus* for each word and proposed several rules that determine the semantic orientations of phrases on the basis of the plus/minus attribute values and the positive/negative attribute values of the component words. For example, a rule [negative+minus=positive] determines "low (*minus*) risk (*negative*)" to be positive. Wilson et al. (2005) worked on phrase-level semantic orientations. They introduced a polarity shifter, which is almost equivalent to the *plus/minus* attribute above. They manually created the list of polarity shifters. The method that we propose in this paper is an automatic version of Inui's or Wilson et al.'s idea, in the sense that the method automatically creates word clusters and their polarity shifters.

## 3 Latent Variable Models for Semantic Orientations of Phrases

As mentioned in the Introduction, the semantic orientation of a phrase is not a mere sum of its component words. If we know that "low risk" is positive, and that "risk" and "mortality", in some sense, belong to the same semantic cluster, we can infer that "low mortality" is also positive. Therefore, we propose to use latent variable models to extract such latent semantic clusters and to realize an accurate classification of phrases (we focus

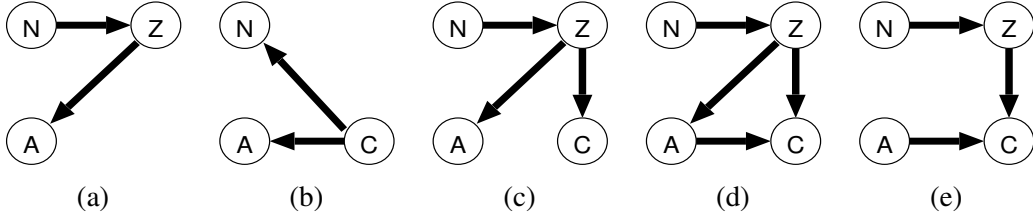


Figure 1: Graphical representations:(a) PLSI, (b) naive bayes, (c) 3-PLSI, (d) triangle, (e) U-shaped; Each node indicates a random variable. Arrows indicate statistical dependency between variables.  $N$ ,  $A$ ,  $Z$  and  $C$  respectively correspond to nouns, adjectives, latent clusters and semantic orientations.

on two-term phrases in this paper). The models adopted in this paper are also used for collaborative filtering by Hofmann (2004).

With these models, the nouns (e.g., “risk” and “mortality”) that become positive by *reducing their degree or amount* would make a cluster. On the other hand, the adjectives or verbs (e.g., “reduce” and “decrease”) that are related to *reduction* would also make a cluster.

Figure 1 shows graphical representations of statistical dependencies of models with a latent variable.  $N$ ,  $A$ ,  $Z$  and  $C$  respectively correspond to nouns, adjectives, latent clusters and semantic orientations. Figure 1-(a) is the PLSI model, which cannot be used in this task due to the absence of a variable for semantic orientations. Figure 1-(b) is the naive bayes model, in which nouns and adjectives are statistically independent of each other given the semantic orientation. Figure 1-(c) is, what we call, *the 3-PLSI model*, which is the 3-observable variable version of the PLSI. We call Figure 1-(d) *the triangle model*, since three of its four variables make a triangle. We call Figure 1-(e) *the U-shaped model*. In the triangle model and the U-shaped model, adjectives directly influence semantic orientations (rating categories) through the probability  $P(c|az)$ . While nouns and adjectives are associated with the same set of clusters  $Z$  in the 3-PLSI and the triangle models, only nouns are clustered in the U-shaped model.

In the following, we construct a probability model for the semantic orientations of phrases using each model of (b) to (e) in Figure 1. We explain in detail the triangle model and the U-shaped model, which we will propose to use for this task.

### 3.1 Triangle Model

Suppose that a set  $D$  of tuples of noun  $n$ , adjective  $a$  (predicate, generally) and the rating  $c$  is given :

$$D = \{(n_1, a_1, c_1), \dots, (n_{|D|}, a_{|D|}, c_{|D|})\}, \quad (1)$$

where  $c \in \{-1, 0, 1\}$ , for example. This can be easily expanded to the case of  $c \in \{1, \dots, 5\}$ . Our purpose is to predict the rating  $c$  for unknown pairs of  $n$  and  $a$ .

According to Figure 1-(d), the generative probability of  $n, a, c, z$  is the following :

$$P(nacz) = P(z|n)P(a|z)P(c|az)P(n). \quad (2)$$

Remember that for the original PLSI model,  $P(naz) = P(z|n)P(a|z)P(n)$ .

We use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to estimate the parameters of the model. According to the theory of the EM algorithm, we can increase the likelihood of the model with latent variables by iteratively increasing the  $Q$ -function. The  $Q$ -function (i.e., the expected log-likelihood of the joint probability of complete data with respect to the conditional posterior of the latent variable) is expressed as :

$$Q(\theta) = \sum_{nac} f_{nac} \sum_z \bar{P}(z|nac) \log P(nazc|\theta), \quad (3)$$

where  $\theta$  denotes the set of the new parameters.  $f_{nac}$  denotes the frequency of a tuple  $n, a, c$  in the data.  $\bar{P}$  represents the posterior computed using the current parameters.

The E-step (expectation step) corresponds to simple posterior computation :

$$\bar{P}(z|nac) = \frac{P(z|n)P(a|z)P(c|az)}{\sum_z P(z|n)P(a|z)P(c|az)}. \quad (4)$$

For derivation of update rules in the M-step (maximization step), we use a simple Lagrange method for this optimization problem with constraints :  $\forall z, \sum_n P(n|z) = 1$ ,  $\forall z, \sum_a P(a|z) = 1$ , and  $\forall a, z, \sum_c P(c|az) = 1$ . We obtain the following update rules :

$$P(z|n) = \frac{\sum_{ac} f_{nac} \bar{P}(z|nac)}{\sum_{ac} f_{nac}}, \quad (5)$$

$$P(y|z) = \frac{\sum_{nac} f_{nac} \bar{P}(z|nac)}{\sum_{nac} f_{nac} \bar{P}(z|nac)}, \quad (6)$$

$$P(c|az) = \frac{\sum_n f_{nac} \bar{P}(z|nac)}{\sum_{nac} f_{nac} \bar{P}(z|nac)}. \quad (7)$$

These steps are iteratively computed until convergence. If the difference of the values of  $Q$ -function before and after an iteration becomes smaller than a threshold, we regard it as converged.

For classification of an unknown pair  $n, a$ , we compare the values of

$$P(c|na) = \frac{\sum_z P(z|n)P(a|z)P(c|az)}{\sum_{cz} P(z|n)P(a|z)P(c|az)}. \quad (8)$$

Then the rating category  $c$  that maximize  $P(c|na)$  is selected.

### 3.2 U-shaped Model

We suppose that the conditional probability of  $c$  and  $z$  given  $n$  and  $a$  is expressed as :

$$P(cz|na) = P(c|az)P(z|n). \quad (9)$$

We compute parameters above using the EM algorithm with the  $Q$ -function :

$$Q(\theta) = \sum_{nac} f_{nac} \sum_z \bar{P}(z|nac) \log P(cz|na, \theta) \quad (10)$$

We obtain the following update rules :

E step

$$\bar{P}(z|nac) = \frac{P(c|az)P(z|n)}{\sum_z P(c|az)P(z|n)}, \quad (11)$$

M step

$$P(c|az) = \frac{\sum_n f_{nac} \bar{P}(z|nac)}{\sum_{nac} f_{nac} \bar{P}(z|nac)}, \quad (12)$$

$$P(z|n) = \frac{\sum_{ac} f_{nac} \bar{P}(z|nac)}{\sum_{ac} f_{nac}}. \quad (13)$$

For classification, we use the formula :

$$P(c|na) = \sum_z P(c|az)P(z|n). \quad (14)$$

### 3.3 Other Models for Comparison

We will also test the 3-PLSI model corresponding to Figure 1-(c).

In addition to the latent models, we test a baseline classifier, which uses the posterior probability :

$$P(c|na) \propto P(n|c)P(a|c)P(c). \quad (15)$$

This baseline model is equivalent to the 2-term naive bayes classifier (Mitchell, 1997). The graphical representation of the naive bayes model is (b) in Figure 1. The parameters are estimated as :

$$P(n|c) = \frac{1 + f_{nc}}{|N| + f_c}, \quad (16)$$

$$P(a|c) = \frac{1 + f_{ac}}{|A| + f_c}, \quad (17)$$

where  $|N|$  and  $|A|$  are the numbers of the words for  $n$  and  $a$ , respectively.

Thus, we have four different models : naive bayes (baseline), 3-PLSI, triangle, and U-shaped.

### 3.4 Discussions on the EM computation, the Models and the Task

In the actual EM computation, we use the tempered EM (Hofmann, 2001) instead of the standard EM explained above, because the tempered EM can avoid an inaccurate estimation of the model caused by ‘‘over-confidence’’ in computing the posterior probabilities. The tempered EM can be realized by a slight modification to the E-step, which results in a new E-step :

$$\bar{P}(z|nac) = \frac{(P(c|az)P(z|n))^\beta}{\sum_z (P(c|az)P(z|n))^\beta}, \quad (18)$$

for the U-shaped model, where  $\beta$  is a positive hyper-parameter, called the *inverse temperature*. The new E-steps for the other models are similarly expressed.

Now we have two hyper-parameters : inverse temperature  $\beta$ , and the number of possible values  $M$  of latent variables. We determine the values of these hyper-parameters by splitting the given training dataset into two datasets (the temporary training dataset 90% and the held-out dataset 10%), and by obtaining the classification accuracy for the held-out dataset, which is yielded by the classifier with the temporary training dataset.

We should also note that  $Z$  (or any variable) should not have incoming arrows simultaneously from  $N$  and  $A$ , because the model with such arrows has  $P(z|na)$ , which usually requires an excessively large memory.

To work with numerical scales of the rating variable (i.e., the difference between  $c = -1$  and  $c = 1$  should be larger than that of  $c = -1$  and  $c = 0$ ), Hofmann (2004) used also a Gaussian distribution for  $P(c|az)$  in collaborative filtering. However, we do not employ a Gaussian, because in our dataset, the number of rating classes is

only 3, which is so small that a Gaussian distribution cannot be a good approximation of the actual probability density function. We conducted preliminary experiments with the model with Gaussians, but failed to obtain good results. For other datasets with more classes, Gaussians might be a good model for  $P(c|az)$ .

The task we address in this paper is somewhat similar to the trigram prediction task, in the sense that both are classification tasks given two words. However, we should note the difference between these two tasks. In our task, the actual answer given two specific words are fixed as illustrated by the fact ‘high+salary’ is always positive, while the answer for the trigram prediction task is randomly distributed. We are therefore interested in the semantic orientations of unseen pairs of words, while the main purpose of the trigram prediction is accurately estimate the probability of (possibly seen) word sequences.

In the proposed models, only the words that appeared in the training dataset can be classified. An attempt to deal with the unseen words is an interesting task. For example, we could extend our models to semi-supervised models by regarding  $C$  as a partially observable variable. We could also use distributional similarity of words (e.g., based on window-size cooccurrence) to find an observed word that is most similar to the given unseen word. However, such methods would not work for the semantic orientation classification, because those methods are designed for simple cooccurrence and cannot distinguish “survival-rate” from “infection-rate”. In fact, the similarity-based method mentioned above failed to work efficiently in our preliminary experiments. To solve the problem of unseen words, we would have to use other linguistic resources such as a thesaurus or a dictionary.

## 4 Experiments

### 4.1 Experimental Settings

We extracted pairs of a noun (subject) and an adjective (predicate), from Mainichi newspaper articles (1995) written in Japanese, and annotated the pairs with semantic orientation tags : positive, neutral or negative. We thus obtained the labeled dataset consisting of 12066 pair instances (7416 different pairs). The dataset contains 4459 negative instances, 4252 neutral instances, and 3355 positive instances. The number of distinct nouns is 4770 and the number of distinct adjectives is 384.

To check the inter-annotator agreement between two annotators, we calculated  $\kappa$  statistics, which was 0.640. This value is allowable, but not quite high. However, positive-negative disagreement is observed for only 0.7% of the data. In other words, this statistics means that the task of extracting neutral examples, which has hardly been explored, is intrinsically difficult.

We employ 10-fold cross-validation to obtain the average value of the classification accuracy. We split the dataset such that there is no overlapping pair (i.e., any pair in the training dataset does not appear in the test dataset).

If either of the two words in a pair in the test dataset does not appear in the training dataset, we excluded the pair from the test dataset since the problem of unknown words is not in the scope of this research. Therefore, we evaluate the pairs that are not in the training dataset, but whose component words appear in the training dataset.

In addition to the original dataset, which we call the *standard* dataset, we prepared another dataset in order to examine the power of the latent variable model. The new dataset, which we call the *hard* dataset, consists only of examples with 17 difficult adjectives such as “high”, “low”, “large”, “small”, “heavy”, and “light”.<sup>1</sup> The semantic orientations of pairs including these difficult words often shift depending on the noun they modify. Thus, the hard dataset is a subset of the standard dataset. The size of the hard dataset is 4787. Please note that the hard dataset is used only as a test dataset. For training, we always use the standard dataset in our experiments.

We performed experiments with all the values of  $\beta$  in  $\{0.1, 0.2, \dots, 1.0\}$  and with all the values of  $M$  in  $\{10, 30, 50, 70, 100, 200, 300, 500\}$ , and predicted the best values of the hyper-parameters with the held-out method in Section 3.4.

### 4.2 Results

The classification accuracies of the four methods with  $\beta$  and  $M$  predicted by the held-out method are shown in Table 1. Please note that the naive bayes method is irrelevant of  $\beta$  and  $M$ . The table shows that the triangle model and the U-shaped

<sup>1</sup>The complete list of the 17 Japanese adjectives with their English counterparts are : takai (high), hikui (low), ookii (large), chiisai (small), omoi (heavy), karui (light), tsuyoi (strong), yowai (weak), ooi (many), sukunai (few/little), nai (no), sugoi (terrific), hageshii (terrific), hukai (deep), asai (shallow), nagai (long), mizikai (short).

Table 1: Accuracies with predicted  $\beta$  and  $M$

	standard			hard		
	accuracy	$\beta$	$M$	accuracy	$\beta$	$M$
Naive Bayes	73.40	–	–	65.93	–	–
3-PLSI	67.02	0.73	91.7	60.51	0.80	87.4
Triangle model	81.39	0.60	174.0	77.95	0.60	191.0
U-shaped model	81.94	0.64	60.0	75.86	0.65	48.3

model achieved high accuracies and outperformed the naive bayes method. This result suggests that we succeeded in capturing the internal structure of semantically oriented phrases by way of latent variables. The more complex structure of the triangle model resulted in the accuracy that is higher than that of the U-shaped model.

The performance of the 3-PLSI method is even worse than the baseline method. This result shows that we should use a model in which adjectives can directly influence the rating category.

Figures 2, 3, 4 show cross-validated accuracy values for various values of  $\beta$ , respectively yielded by the 3-PLSI model, the triangle model and the U-shaped model with different numbers  $M$  of possible states for the latent variable. As the figures show, the classification performance is sensitive to the value of  $\beta$ .  $M = 100$  and  $M = 300$  are mostly better than  $M = 10$ . However, this is a tradeoff between classification performance and training time, since large values of  $M$  demand heavy computation. In that sense, the U-shaped model is useful in many practical situations, since it achieved a good accuracy even with a relatively small  $M$ .

To observe the overall tendency of errors, we show the contingency table of classification by the U-shaped model with the predicted values of hyperparameters, in Table 2. As this table shows, most of the errors are caused by the difficulty of classifying *neutral* examples. Only 2.26% of the errors are mix-ups of the *positive* orientation and the *negative* orientation.

We next investigate the causes of errors by observing those mix-ups of the *positive* orientation and the *negative* orientation.

One type of frequent errors is illustrated by the pair “food (’s price) is high”, in which the word “price” is omitted in the actual example<sup>2</sup>. As in this expression, the attribute (price, in this case) of an example is sometimes omitted or not correctly

<sup>2</sup>This kind of ellipsis often occurs in Japanese.

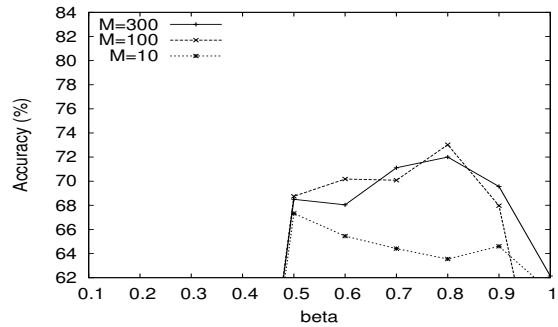


Figure 2: 3-PLSI model with standard dataset

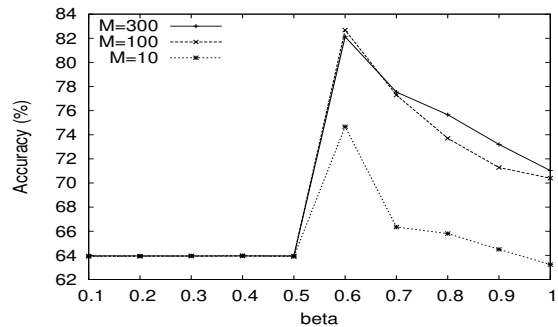


Figure 3: Triangle model with standard dataset

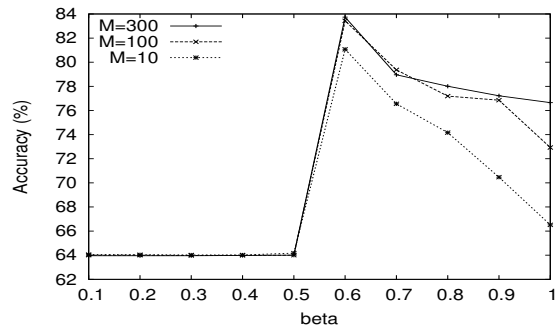


Figure 4: U-shaped model with standard dataset

Table 2: Contingency table of classification result by the U-shaped model

		U-shaped model			
		positive	neutral	negative	sum
Gold standard	positive	1856	281	69	2206
	neutral	202	2021	394	2617
	negative	102	321	2335	2758
	sum	2160	2623	2798	7581

identified. To tackle these examples, we will need methods for correctly identifying attributes and objects. Some researchers are starting to work on this problem (e.g., Popescu and Etzioni (2005)).

We succeeded in addressing the data-sparseness problem by introducing a latent variable. However, this problem still causes some errors. Precise statistics cannot be obtained for infrequent words. This problem will be solved by incorporating other resources such as thesaurus or a dictionary, or combining our method with other methods using external wider contexts (Suzuki et al., 2006; Turney, 2002; Baron and Hirst, 2004).

### 4.3 Examples of Obtained Clusters

Next, we qualitatively evaluate the proposed methods. For several clusters  $z$ , we extract the words that occur more than twice in the whole dataset and are in top 50 according to  $P(z|n)$ . The model used here as an example is the U-shaped model. The experimental settings are  $\beta = 0.6$  and  $M = 60$ . Although some elements of clusters are composed of multiple words in English, the original Japanese counterparts are single words.

- Cluster 1 trouble, objection, disease, complaint, anxiety, anamnesis, relapse
- Cluster 2 risk, mortality, infection rate, onset rate
- Cluster 3 bond, opinion, love, meaning, longing, will
- Cluster 4 vote, application, topic, supporter
- Cluster 5 abuse, deterioration, shock, impact, burden
- Cluster 6 deterioration, discrimination, load, abuse
- Cluster 7 relative importance, degree of influence, number, weight, sense of belonging, wave, reputation

These obtained clusters match our intuition. For example, in cluster 2 are the nouns that are negative when combined with “high”, and positive when combined with “low”. In fact, the posterior probabilities of semantic orientations for cluster 2 are as follows :

$$P(\text{negative}|\text{high}, \text{cluster 2}) = 0.995,$$

$$P(\text{positive}|\text{low}, \text{cluster 2}) = 0.973.$$

With conventional clustering methods based on

the cooccurrence of two words, cluster 2 would include the words resulting in the opposite orientation, such as “success rate”. We succeeded in obtaining the clusters that are suitable for our task, by incorporating the new variable  $c$  for semantic orientation in the EM computation.

## 5 Conclusion

We proposed models for phrases with semantic orientations as well as a classification method based on the models. We introduced a latent variable into the models to capture the properties of phrases. Through experiments, we showed that the proposed latent variable models work well in the classification of semantic orientations of phrases and achieved nearly 82% classification accuracy. We should also note that our method is language-independent although evaluation was on a Japanese dataset.

We plan next to adopt a semi-supervised learning method in order to correctly classify phrases with infrequent words, as mentioned in Section 4.2. We would also like to extend our method to 3- or more term phrases. We can also use the obtained latent variables as features for another classifier, as Fujita et al. (2004) used latent variables of PLSI for the  $k$ -nearest neighbors method. One important and promising task would be the use of semantic orientations of words for phrase level classification.

## References

- Faye Baron and Graeme Hirst. 2004. Collocations as cues to semantic orientation. In *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*.
- Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38.

- Atsushi Fujita, Kentaro Inui, and Yuji Matsumoto. 2004. Detection of incorrect case assignments in automatically generated paraphrases of Japanese sentences. In *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP)*, pages 14–21.
- Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and the Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181.
- Thomas Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196.
- Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22:89–115.
- Takashi Inui. 2004. *Acquiring Causal Knowledge from Text Using Connective Markers*. Ph.D. thesis, Graduate School of Information Science, Nara Institute of Science and Technology.
- Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten de Rijke. 2004. Using wordnet to measure semantic orientation of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, volume IV, pages 1115–1118.
- Nozomi Kobayashi, Takashi Inui, and Kentaro Inui. 2001. Dictionary-based acquisition of the lexical knowledge for p/n analysis (in Japanese). In *Proceedings of Japanese Society for Artificial Intelligence, SLUD-33*, pages 45–50.
- Mainichi. 1995. Mainichi Shimbun CD-ROM version.
- Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. 2005. Sentiment classification using word sub-sequences and dependency sub-trees. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-05)*, pages 301–310.
- Tom M. Mitchell. 1997. *Machine Learning*. McGraw Hill.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'02)*, pages 79–86.
- Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of joint conference on Human Language Technology / Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP'05)*, pages 339–346.
- Frank Z. Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.
- Yasuhiro Suzuki, Hiroya Takamura, and Manabu Okumura. 2006. Application of semi-supervised learning to evaluative expression classification. In *Proceedings of the 7th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-06)*, pages 502–513.
- Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 133–140.
- Kentaro Torisawa. 2001. An unsupervised method for canonicalization of Japanese postpositions. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium (NLPRS 2001)*, pages 211–218.
- Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.
- Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 417–424.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of joint conference on Human Language Technology / Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP'05)*, pages 347–354.