Using Hidden Markov Random Fields to Combine Distributional and Pattern-based Word Clustering

Nobuhiro Kaji and **Masaru Kitsuregawa**
Institute of Industrial Science, University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505 Japan
{kaji,kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract

Word clustering is a conventional and important NLP task, and the literature has suggested two kinds of approaches to this problem. One is based on the distributional similarity and the other relies on the co-occurrence of two words in lexico-syntactic patterns. Although the two methods have been discussed separately, it is promising to combine them since they are complementary with each other. This paper proposes to integrate them using hidden Markov random fields and demonstrates its effectiveness through experiments.

1 Introduction

Word clustering is a technique of grouping similar words together, and it is important for various NLP systems. Applications of word clustering include language modeling (Brown et al., 1992), text classification (Baker and McCallum, 1998), thesaurus construction (Lin, 1998) and so on. Furthermore, recent studies revealed that word clustering is useful for semi-supervised learning in NLP (Miller et al., 2004; Li and McCallum, 2005; Kazama and Torisawa, 2008; Koo et al., 2008).

A well-known approach to grouping similar words is to use distribution of contexts in which target words appear. It is founded on the hypothesis that similar words tend to appear in similar contexts (Harris, 1968). Based on this idea, some studies proposed probabilistic models for word clustering (Pereira et al., 1993; Li and Abe, 1998; Rooth et al., 1999; Torisawa, 2002). Others proposed distributional similarity measures between words (Hindle, 1990; Lin, 1998; Lee, 1999; Weeds et al., 2004). Once such similarity is defined, it is trivial to perform clustering.

On the other hand, some researchers utilized co-occurrence for word clustering. The idea behind it is that similar words tend to co-occur in certain patterns. Considerable efforts have been devoted to measure word similarity based on co-occurrence frequency of two words in a window (Church and Hanks, 1989; Turney, 2001; Terra and Clarke, 2003; Matsuo et al., 2006). In addition to the classical window-based technique, some studies investigated the use of lexico-syntactic patterns (e.g., X or Y) to get more accurate co-occurrence statistics (Chilovski and Pantel, 2004; Bollegala et al., 2007).

These two approaches are complementary with each other, because they are founded on different hypotheses and utilize different corpus statistics. Consider to cluster a set of words based on the distributional similarity. It is likely that some words are difficult to cluster due to the data sparseness or some other problems, while we can still expect that those words are correctly classified using patterns.

This consideration leads us to combine distributional and pattern-based word clustering. In this paper we propose to combine them using mixture models based on hidden Markov random fields. This model was originally proposed by (Basu et al., 2004) for semi-supervised clustering. In semi-supervised clustering, the system is provided with supervision in the form of pair-wise constraints specifying data points that are likely to belong to the same cluster. These constraints are directly incorporated into the clustering process as a prior knowledge. Our idea is to view the co-occurrence

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 401–408
Manchester, August 2008

of two words in lexico-syntactic patterns as constraints, and incorporate them into distributional word clustering.

In summary, this paper discusses the problem of integrating multiple approaches for word clustering. We consider that the clustering results are improved if multiple approaches are successfully combined and if they are complementary with each other. Our contribution is to provide a probabilistic framework for this problem. Although our proposal aims at combining the distributional and pattern-based approaches, it is also applicable to combine other approaches like (Lin et al., 2003), as we will discuss in Section 5.4.

2 Distributional Clustering

This and next section describe distributional and pattern-based word clustering respectively. Section 4 will explain how to combine them.

2.1 Probabilistic model

In distributional word clustering, similarity between words (= nouns) is measured by the distribution of contexts in which they appear. As a context, verbs that appear in certain grammatical relations with the target nouns are typically used. Using the distribution of such verbs, we can express a noun n by a feature vector $\boldsymbol{\phi}(n)$:

$$\boldsymbol{\phi}(n) = (f_{nv_1}, f_{nv_2}, ...f_{nv_V})$$

where f_{nv_i} denotes the frequency of noun-verb pair (n, v_i), and V denotes the number of distinct verbs. The basic idea of using the distribution for clustering is to group n and n' together if $\boldsymbol{\phi}(n)$ and $\boldsymbol{\phi}(n')$ are similar.

Let us consider a soft clustering model. We hypothesize that $\boldsymbol{\phi}(n)$ is a mixture of multinomial, and the probability of n is defined by[1]

$$\begin{aligned} p(n) &= \sum_{z=1}^{Z} p(z)p(\boldsymbol{\phi}(n)|z) \\ &= \sum_{z=1}^{Z} \pi_z \frac{f_n!}{\prod_v f_{nv}!} \prod_v \mu_{vz}^{f_{nv}} \end{aligned}$$

where Z is the number of mixture components, π_z is the mixing coefficient ($\sum_z \pi_z = 1$), $f_n = \sum_v f_{nv}$ is the total number of occurrence of n, and μ_{vz} is the parameter of the multinomial distribution ($\sum_v \mu_{vz} = 1$). In this model the hidden variables can be interpreted as semantic class of nouns.

Now consider a set of nouns $\mathbf{n} = \{n_i\}_{i=1}^{N}$. Let $\mathbf{z} = \{z_i\}_{i=1}^{N}$ be a set of hidden variables corresponding to $\mathbf{n}$. Assuming that the hidden variables are independent and n_i is also independent of other nouns given the hidden variables, the probability of $\mathbf{n}$ is defined by

$$p(\mathbf{n}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{n}|\mathbf{z})$$

where

$$\begin{aligned} p(\mathbf{z}) &= \prod_{i=1}^{N} p(z_i) \\ p(\mathbf{n}|\mathbf{z}) &= \prod_{i=1}^{N} p(n_i|z_i). \end{aligned}$$

Hereafter, we use $p(n|z)$ instead of $p(\boldsymbol{\phi}(n)|z)$ to keep the notation simple. $p(\mathbf{n}|\mathbf{z})$ is the conditional distribution on all nouns given all the hidden variables, and $p(\mathbf{z})$ is the prior distribution on the hidden variables. Computing the log-likelihood of the complete data $(\mathbf{n}, \mathbf{z})$, we found

$$\log p(\mathbf{n}, \mathbf{z}) = \sum_{i=1}^{N} \log p(z_i)p(n_i|z_i). \quad (1)$$

2.2 Parameter estimation

The parameters can be estimated by the EM algorithm. In the E-step, $p(z_i|n_i)$ is computed based on current parameters. It is computed by

$$\begin{aligned} p(z_i = k|n_i) &= \frac{p(z_i = k)p(n_i|z_i = k)}{\sum_z p(z)p(n_i|z)} \\ &= \frac{\pi_k \prod_v \mu_{vk}^{f_{n_i v}}}{\sum_z \pi_z \prod_v \mu_{vz}^{f_{n_i v}}}. \end{aligned}$$

In the M-step, the parameters are re-estimated by using the result of the E-step:

$$\begin{aligned} \mu_{\gamma k} &= \frac{\alpha + \sum_i f_{n_i \gamma} p(z_i = k|n_i)}{\alpha V + \sum_v \sum_i f_{n_i v} p(z_i = k|n_i)} \\ \pi_k &= \frac{\alpha + \sum_i p(z_i = k|n_i)}{\alpha Z + \sum_z \sum_i p(z_i = z|n_i)} \end{aligned}$$

where α is a smoothing factor.[2] Both steps are repeated until a convergence criteria is satisfied. The important point to note is that the E-step can be computed using the above equation because the hidden variables are independent.

[1] We ignored $p(f_n)$ by assuming that it is independent of hidden variables. See (McCallum and Nigam, 1998) for detail discussion.

[2] α=1.0 in our experiment.

X *ya* Y	X *mo* Y *mo*	X *to* Y *to*	X, Y *nado*
(X or Y)	(X and Y)	(X and Y)	(X, Y etc.)

Table 1: Four lexico-syntactic patterns, where X and Y are extracted as co-occurring words. Note that *ya*, *mo*, and *to* are Japanese postpositions, and they correspond to *or* or *and* in English.

3 Pattern-based Clustering

A graph-based algorithm was employed in order to cluster words using patterns.

3.1 Graph Construction

We first construct the graph in which vertices and edges correspond to words and their co-occurrences in patterns respectively (Figure 1). We employed four lexico-syntactic patterns (Table 1) to extract co-occurrence of two words from corpus. Note that we target Japanese in this paper although our proposal is independent of languages. The edges are weighted by the strength of co-occurrence that is computed by the Point-wise Mutual Information (PMI):

$$\text{PMI}(n_i, n_j) \quad = \quad \log \frac{f(n_i, n_j) f(*, *)}{f(n_i, *) f(*, n_j)}$$

where $f(n_i, n_j)$ is the co-occurrence frequency of two nouns, and '$*$' means summation over all nouns. If PMI is less than zero, the edge is removed.

3.2 Graph Partitioning

Assuming that similar words tend to co-occur in the lexico-syntactic patterns, it is reasonable to consider that a dense subgraph is a good cluster (Figure 1). Following (Matsuo et al., 2006), we exploit the Newman clustering (Newman, 2004) to partition the graph into such dense subgraphs.

We start by describing Newman's algorithm for unweighted graphs and we will generalize it to weighted graphs later. The Newman clustering is an algorithm that divides a graph into subgraphs based on connectivity. Roughly speaking, it divides a graph such that there are a lot of edges between vertices in the same cluster. In the algorithm goodness of clustering is measured by score Q:

$$Q = \sum_i \Big(e_{ii} - a_i^2\Big)$$

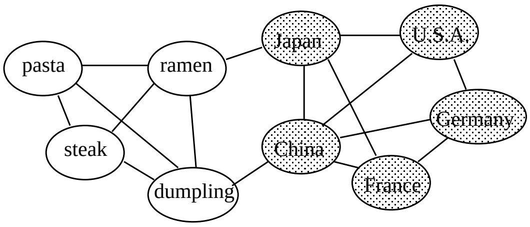

Figure 1: An example of the graph consisting of two dense subgraphs.

where

$$\begin{aligned} e_{ij} &= \frac{\text{\# of edges between two vertices in cluster } i \text{ and } j}{\text{\# of all edges}} \\ a_i &= \sum_k e_{ik}. \end{aligned}$$

The term e_{ij} is the fraction of edges between cluster i and j. a_i is the sum of e_{ik} over all clusters, and a_i^2 represents the expected number of fraction of edges within the cluster i when edges are given at random. See (Newman, 2004) for the detail.

The Newman clustering optimizes Q in an agglomerative fashion. At the beginning of the algorithm every vertex forms a singleton cluster, and we repeatedly merge two clusters so that the join results in the largest increase in Q. The change in Q when cluster i and j are merged is given by

$$\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j).$$

The above procedure is repeated until Q reaches local maximum.

The algorithm can be easily generalized to weighted graphs by substituting "sum of weights of edges" for "# of edges" in the definition of e_{ij}. The other part of the algorithm remains the same.

4 Integration based on Hidden Markov Random Fields

This section represents how to integrate the distribution and pattern for word clustering.

4.1 Background and idea

Clustering has long been discussed as an unsupervised learning problem. In some applications, however, it is possible to provide some form of supervision by hand in order to improve the clustering result. This motivated researchers to investigate semi-supervised clustering, which uses not only unlabeled data but supervision in the form of pair-wise constraints (Basu et al., 2004). In this framework, the clustering system is provided with a set of pair-wise constraints specifying data points that are likely to belong to the same cluster. These constraints are directly incorporated into the clustering process as a prior knowledge.

Our idea is to view the co-occurrence of two words in lexico-syntactic patterns as constraints, and incorporate them into the distributional clustering. The rest of this section describes how to extend the distributional clustering so as to incorporate the constraints, and how to generate the constraints using the patterns.

4.2 Probabilistic model

Let $\mathbb{C}$ be a set of pair-wise constraints, and consider to incorporate the constraints into the distributional clustering (Section 2). In what follows we assume each constraint $\langle i, j \rangle \in \mathbb{C}$ represents that z_i and z_j are likely to have the same value, and it is associated with a weight $w_{ij} (> 0)$ corresponding to a penalty for constraint violation.

It is easy to extend the distributional clustering algorithm so as to incorporate the constraints. This is done by just changing the prior distribution on hidden variables $p(\mathbf{z})$. Following (Basu et al., 2004), we construct the Markov random field on the hidden variables so as to incorporate the constraints. The new prior distribution is defined as

$$p(\mathbf{z}) = \prod_{i=1}^{N} p(z_i) \times \frac{1}{G} \exp\{- \sum_{\langle i,j \rangle \in \mathbb{C}} \delta(z_i \neq z_j) w_{ij}\}$$

where $\delta(\cdot)$ is the delta function. $\delta(z_i \neq z_j)$ takes one if the constraint $\langle i, j \rangle$ is violated and otherwise zero. G is the normalization factor of the Markov random field (the second term).

By examining the log-likelihood of the complete data, we can see how violation of constraints is penalized. Using the new prior distribution, we get

$$\begin{aligned}\log p(\mathbf{n}, \mathbf{z}) &= \sum_{i=1}^{N} \log p(z_i) p(n_i | z_i) \\ &- \sum_{\langle i,j \rangle \in \mathbb{C}} \delta(z_i \neq z_j) w_{ij} \\ &- \log G.\end{aligned}$$

The first term in the right-hand side is equal to the log-likelihood of the multinomial mixture, namely equation (1). The second term can be interpreted as the penalty for constraint violation. The last term is a constant.

It is worth pointing out that the resulting algorithm makes a soft assignment and polysemous words can belong to more than one clusters.

4.3 Parameter estimation

The parameters are estimated by the EM algorithm. The M-step is exactly the same as discussed in Section 2.2. The problem is that the hidden variables are no longer independent and the E-step requires the calculation of

$$\begin{aligned}p(z_i | \mathbf{n}) &= \sum_{\mathbf{z}_{\neg i}} p(\mathbf{z}_{\neg i}, z_i | \mathbf{n}) \\ &\propto \sum_{\mathbf{z}_{\neg i}} p(\mathbf{z}_{\neg i}, z_i) p(\mathbf{n} | \mathbf{z}_{\neg i}, z_i)\end{aligned}$$

where $\mathbf{z}_{\neg i}$ means all hidden variables but z_i. The computation of the above equation is intractable because the summation in it requires $O(Z^{N-1})$ operations.

Instead of exactly computing $p(z_i | \mathbf{n})$, we approximate it by using the mean field approximation (Lange et al., 2005). In the mean field approximation, $p(\mathbf{z} | \mathbf{n})$ is approximated by a factorized distribution $q(\mathbf{z})$, in which all hidden variables are independent:

$$q(\mathbf{z}) = \prod_{i=1}^{N} q_i(z_i). \quad (2)$$

Using $q(\mathbf{z})$ instead of $p(\mathbf{z} | \mathbf{n})$, computation of the E-step can be written as follows:

$$p(z_i | \mathbf{n}) \simeq \sum_{\mathbf{z}_{\neg i}} q(\mathbf{z}_{\neg i}, z_i) = q_i(z_i). \quad (3)$$

The parameters of $q(\mathbf{z})$ are determined such that the KL divergence between $q(\mathbf{z})$ and $p(\mathbf{z} | \mathbf{n})$ is minimized. In other words, the approximate distribution $q(\mathbf{z})$ is determined by minimizing

$$\sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z} | \mathbf{n})} \quad (4)$$

under the condition that $\sum_k q_i(z_i = k) = 1$ for all i. This optimization problem can be resolved by introducing Lagrange multipliers. Because we cannot get the solution in closed form, an iterative method is employed. Taking the derivative of equation (4) with respect to a parameter $q_{ik} = q_i(z_i = k)$ and setting it to zero, we get the following updating formula:

$$q_{ik}^{(t+1)} \propto p(n_i, k) \exp\{- \sum_{j \in \mathbb{N}_i} (1 - q_{jk}^{(t)}) w_{ij}\} \quad (5)$$

where $\mathcal{N}_i = \{j|\langle i, j\rangle \in \mathbb{C}\}$ and $q_{ik}^{(t)}$ is the value of q_{ik} at t-th iteration. The derivation of this formula is found in Appendix.

4.4 Generation of constraints

It is often pointed out that even small amounts of misspecified constraints significantly decrease the performance of semi-supervised clustering. This is because the error of misspecified constraints is propagated to the entire transitive neighborhoods of the constrained data (Nelson and Cohen, 2007). As an example, consider that we have two constraints $\langle i, j\rangle$ and $\langle j, k\rangle$. If the former is misspecified one, the error propagate to k through j.

To tackle this problem, we propose a technique to put an upper bound θ on the size of the transitive neighborhoods. Our constraint generation process is as follows. To begin with, we modified the Newman clustering so that the maximum cluster size does not exceed θ. This can be done by prohibiting such merge that results in larger cluster than θ. Given the result of the modified Newman clustering, it is straightforward to generate constraints. Constraints are generated between two nouns in the same cluster if they co-occur in the lexico-syntactic patterns at least one time. The penalty for constraint violation w_{ij} was set to PMI(n_i, n_j). This procedure obviously ensures that the size of the transitive neighborhoods is less than θ.

5 Experiments

5.1 Data sets

We parsed 15 years of news articles by KNP[3] so as to obtain data sets for the distributional and pattern-based word clustering (Table 2). The number of distinct nouns in total was 297,719. Note that, due to the computational efficiency, we removed such nouns that appeared less than 10 times with verbs and did not appear at all in the patterns.

A test set was created using manually tailored Japanese thesaurus (Ikehara et al., 1997). We randomly selected 500 unambiguous nouns from 25 categories (20 words for each category).

5.2 Baselines

For comparison we implemented the following baseline systems.

- The multinomial mixture (Section 2).
- The Newman clustering (Newman, 2004).
- Three K-means algorithms using different distributional similarity or dissimilarity measures: cosine, α-skew divergence (Lee, 1999)[4], and Lin's similarity (Lin, 1998).
- The CBC algorithm (Lin and Pantel, 2002; Pantel and Lin, 2002).

nouns	208,934
verbs	64,954
noun-verb pairs	4,804,715
nouns	245,465
noun-noun pairs	633,302

Table 2: Data sets statistics. The first and second row shows the number of distinct words (and word pairs) used for the distributional and pattern-based word clustering respectively.

5.3 Evaluation procedure

All the nouns in the data set were clustered by the proposed and baseline systems.[5] For the mixture models and K-means, the number of clusters was set to 1,000. The parameter θ was set to 100.

The result was assessed by precision and recall using the test data. The precision and recall were computed by the B-CUBED algorithm as follows (Bagga and Baldwin, 1998). For each noun n_i in the test data, precision$_i$ and recall$_i$ are defined as

$$\text{precision}_i = \frac{|S_i \cap T_i|}{|S_i|}$$

$$\text{recall}_i = \frac{|S_i \cap T_i|}{|T_i|}$$

where S_i is the system generated cluster containing n_i and T_i is the goldstandard cluster containing n_i. The precision and recall are defined as an average of precision$_i$ and recall$_i$ for all the nouns in the test data respectively. The result of soft clustering models cannot be directly evaluated by the precision and recall. In such cases, each noun is assigned to the cluster that maximizes $p(z|n)$.

5.4 The result and discussion

Table 3 shows the experimental results. The best results for each statistic are shown in bold. For the mixture models and K-means, the precision and recall are an average of 10 trials.

Table 3 demonstrates the impact of combining distribution and pattern. Our method outperformed

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

[4] α = 0.99 in our experiment.

[5] Our implementation is available from http://www.tkl.iis.u-tokyo.ac.jp/~kaji/clustering.

	P	R	F_1
proposed	.383	**.437**	**.408**
multinomial mixture	.360	.374	.367
Newman (2004)	.318	.353	.334
cosine	.603	.114	.192
α-skew divergence (Lee, 1999)	.730	.155	.255
Lin's similarity (Lin, 1998)	.691	.096	.169
CBC (Lin and Pantel, 2002)	**.981**	.060	.114

Table 3: Precision, recall, and F-measure.

	$f_n = 0$			$f_n \neq 0$		
	P	R	F_1	P	R	F_1
proposed	**.320**	.632	**.435**	**.412**	**.450**	**.430**
multi.	.099	**1.000**	.181	.402	.394	.398

Table 4: Detail comparison with the multinomial mixture.

	$f(n_i, *) = 0$			$f(n_i, *) \neq 0$		
	P	R	F_1	P	R	F_1
proposed	**.600**	.456	**.518**	**.380**	**.479**	**.424**
Newman	.071	**1.000**	.133	.354	.412	.381

Table 5: Detail comparison with the Newman clustering.

all the baseline systems. It was statistically significantly better than the multinomial mixture (P $<$ 0.01, Mann-Whitney U-test). Note that it is possible to improve some baseline systems, especially CBC, by tuning the parameters. For CBC we simply used the same parameter values as reported in (Lin and Pantel, 2002).

Compared with the multinomial mixture, one advantage of our method is that it has broad coverage. Our method can successfully handle *unknown words*, which do not appear with verbs at all (i.e., $f_n = 0$ and $\boldsymbol{\phi}(n)$ is zero vector), if they co-occur with other words in the lexico-syntactic patterns. For unknown words, the hidden variables are determined based only on $p(\mathbf{z})$ because $p(n|z)$ takes the same value for all hidden variables. This means that our method clusters unknown words using pair-wise constraints. On the other hand, the multinomial mixture assigns all the unknown words to the cluster that maximizes $p(z)$.

The test set included 51 unknown words.[6] We split the test set into two parts: $f_n = 0$ and $f_n \neq 0$, and calculated precision and recall for each subset (Table 4). Although the improvement is especially significant for the unknown words, we can clearly confirm the improvement for both subsets. For the Newman clustering we can discuss similar things (Table 5). Different from the Newman clustering, our method can handle nouns that do not co-occur with other nouns if $0 < f_n$. In this case the test set included 64 unknown words.

It is interesting to point out that our framework can further incorporate lexico-syntactic patterns for dissimilar words (Lin et al., 2003). Namely, we can use patterns so as to prevent distributionally similar but semantically different words (e.g., ally and supporter (Lin et al., 2003)) from being assigned to the same cluster. This can be achieved by using cannot-link constraints, which specify data points that are likely to belong to different clusters (Basu et al., 2004). The remaining problem is which patterns to use so as to extract dissimilar words. Although this problem has already been discussed by (Lin et al., 2003), they mainly addressed antonyms. We believe that a more exhaustive investigation is required. In addition, it is still unclear whether dissimilar words are really useful to improve clustering results.

One problem that we did not examine is how to determine optimal number of clusters. In the experiment, the number was decided with trial-and-error through our initial experiment. We leave it as our future work to test methods of automatically determining the cluster number (Pedersen and Kulkarni, 2006; Blei and Jordan, 2006).

6 Related work

As far as we know, the distributional and pattern-based word clustering have been discussed independently (e.g., (Pazienza et al., 2006)). One of the most relevant work is (Bollegala et al., 2007), which proposed to integrate various patterns in order to measure semantic similarity between words. Although they extensively discussed the use of patterns, they did not address the distributional approach.

Mirkin (2006) pointed out the importance of integrating distributional similarity and lexico-syntactic patterns, and showed how to combine the two approaches for textual entailment acquisition. Although their work inspired our research, we discussed word clustering, which is related to but different from entailment acquisition.

Lin (2003) also proposed to use both distributional similarity and lexico-syntactic patterns for finding synonyms. However, they present an opposite viewpoint from our research. Their proposal is to exploit patterns in order to filter dissimilar

[6] The baseline systems assigned the unknown words to a default cluster as the multinomial mixture does.

words. As we have already discussed, the integration of such patterns can also be formalized using similar probabilistic model to ours.

A variety of studies discussed determining polarity of words. Because this problem is ternary (positive, negative, and neutral) classification of words, it can be seen as one kind of word clustering. The literature suggested two methods of determining polarity, and they are analogous to the distributional and co-occurrence-based approaches in word clustering (Takamura et al., 2005; Higashiyama et al., 2008). We consider it is also promising to integrate them for polarity determination.

7 Conclusion

The distributional and pattern-based word clustering have long been discussed separately despite the potentiality for their integration. In this paper, we provided a probabilistic framework for combining the two approaches, and demonstrated that the clustering result is significantly improved.

Our important future work is to extend current framework so as to incorporate patterns for dissimilar words using cannot-link constraints. We consider such patterns further improve the clustering result.

Combining distribution and pattern is important for other NLP problems as well (e.g., entailment acquisition, polarity determination). Although this paper examined word clustering, we consider a part of our idea can be applied to other problems.

Acknowledgement

This work was supported by the *Comprehensive Development of e-Society Foundation Software* program of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

Bagga, Amit and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of ACL*, pages 79–85.

Baker, L. Douglas and Andrew Kachites McCallum. 1998. Distributional clustering of words for text classification. In *Proceedings of SIGIR*, pages 96–103.

Basu, Sugato, Mikhail Bilenko, and Raymond J. Mooney. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of SIGKDD*, pages 59–68.

Blei, David M. and Michael I. Jordan. 2006. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144.

Bollegala, Danushka, Yutaka Matsuo, and Mitsuru Ishizuka. 2007. An integrated approach to measuring semantic similarity between words using information available on the web. In *Proceedings of NAACL*, pages 340–347.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Rober L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Chilovski, Timothy and Patrick Pantel. 2004. VERBOCEAN: Mining the web for fine-grained semantic verb relations. In *Proceedings of EMNLP*, pages 33–40.

Church, Kenneth Ward and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of ACL*, pages 76–83.

Harris, Zellig. 1968. *Mathematical Structure of Language*. New York: Wiley.

Higashiyama, Masahiko, Kentaro Inui, and Yuji Matsumoto. 2008. Learning polarity of nouns by selectional preferences of predicates (in Japanese). In *Proceedings of the Association for NLP*, pages 584–587.

Hindle, Donald. 1990. Noun classification from predicate-argument structure. In *Proceedings of ACL*, pages 268–275.

Ikehara, Satoru, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentarou Ogura, and Yoshifumi Oyama Yoshihiko Hayashi, editors. 1997. *Japanese Lexicon*. Iwanami Publishing.

Kazama, Jun'ichi and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proceedings of ACL*, pages 407–415.

Koo, Terry, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL*, pages 595–603.

Lange, Tilman, Martin H.C. Law, Anil K. Jain, and Joachim M. Buhmann. 2005. Learning with constrained and unlabelled data. In *Proceedings of CVPR*, pages 731–738.

Lee, Lillian. 1999. Measures of distributional similarity. In *Proceedings of ACL*, pages 25–32.

Li, Hang and Naoki Abe. 1998. Word clustering and disambiguation based on co-occurrence. In *Proceedings of ACL-COLING*, pages 749–755.

Li, Wei and Andrew McCallum. 2005. Semi-supervised sequence modeling with syntactic topic models. In *Proceedings of AAAI*, pages 813–818.

Lin, Dekang and Patrick Pantel. 2002. Concept discovery from text. In *Proceeodings of COLING*, pages 577–583.

Lin, Dekang, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of IJCAI*, pages 1492–1493.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of ACL-COLING*, pages 768–774.

Matsuo, Yutaka, Takeshi Sakaki, Koki Uchiyama, and Mitsuru Ishizuka. 2006. Graph-based word clustering using a web search engine. In *Proceedings of EMNLP*, pages 542–550.

McCallum, Andrew and Kamal Nigam. 1998. A comparison of event models for naive Bayes text classification. In *Proceedings of AAAI Workshop on Learning for Text Categorization*, pages 41–48.

Miller, Scott, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of NAACL*, pages 579–586.

Mirkin, Shachar, Ido Dagan, and Maayan Geffet. 2006. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *Proceedings of COLING-ACL Poster Sessions*, pages 579–586.

Nelson, Blaine and Ira Cohen. 2007. Revisiting probabilistic models for clustering with pair-wise constraints. In *Proceedings of ICML*, pages 673–680.

Newman, Mark. 2004. Fast algorithm for detecting community structure in networks. In *Phys. Rev. E 69*.

Pantel, Patrick and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of SIGKDD*, pages 613–619.

Pazienza, Maria Teresa, Marco Pennacchiotti, and Fabio Massimo Zanzotto. 2006. Discovering verb relations in corpora: Distributional versus non-distributional approaches. In *Proceedings of IEA/AIE*, pages 1042–1052.

Pedersen, Ted and Anagha Kulkarni. 2006. Automatic cluster stopping with criterion functions and the gap statistic. In *Proceedings of HLT/NAACL, Companion Volume*, pages 276–279.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Proceedings of ACL*, pages 183–190.

Rooth, Mats, Stefan Riezler, Detlef Prescher, Glenn Garrroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of ACL*, pages 104–111.

Takamura, Hiroya, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of ACL*, pages 133–140.

Terra, Egidio and C.L.A. Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Proceedings of NAACL*, pages 165–172.

Torisawa, Kentaro. 2002. An unsupervised learning method for associative relationships between verb phrases. In *Proceedings of COLING*, pages 1009–1015.

Turney, Peter. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of ECML*, pages 491–502.

Weeds, Julie, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of COLING*, pages 1015–1021.

Appendix. Derivation of the updating formula

We can rewrite equation (4) as follows:

$$\begin{aligned}
(4) \quad &= \sum_{\mathbf{z}} q(\mathbf{z}) \log q(\mathbf{z}) && (6)\\
&- \sum_{\mathbf{z}} q(\mathbf{z}) \sum_{i=1}^{N} \log p(n_i, z_i) && (7)\\
&+ \sum_{\mathbf{z}} q(\mathbf{z}) \sum_{\langle i,j\rangle \in \mathbb{C}} \delta(z_i \neq z_j) w_{ij} && (8)\\
&+ \text{const} && (9)
\end{aligned}$$

where we made use of the fact that $\log p(\mathbf{z}|\mathbf{n}) = \log p(\mathbf{n}|\mathbf{z})p(\mathbf{z}) + \text{const}$. Taking the derivative of equation (6), (7), and (8) with respect to q_{ik}, we found

$$\begin{aligned}
\frac{\partial(6)}{\partial q_{ik}} &= \log q_{ik} + \text{const}\\
\frac{\partial(7)}{\partial q_{ik}} &= -\log p(n_i, k) + \text{const}\\
\frac{\partial(8)}{\partial q_{ik}} &= \sum_{\mathbf{z}_{\neg i}} q(\mathbf{z}_{\neg i}) \sum_{j \in \mathcal{N}_i} \delta(z_j \neq k) w_{ij} + \text{const}\\
&= \sum_{j \in \mathcal{N}_i} \sum_{\mathbf{z}_{\neg i}} q(\mathbf{z}_{\neg i}) \delta(z_j \neq k) w_{ij} + \text{const}\\
&= \sum_{j \in \mathcal{N}_i} (1 - q_{jk}) w_{ij} + \text{const}
\end{aligned}$$

where const denotes terms independent of k. Making use of these results, the updating formula can be derived by taking the derivative of equation (4) with respect to q_{ik} and setting it to zero.