Natural Language Engineering **8** (1): 25–42. © 2002 Cambridge University Press
DOI: 10.1017/S1351324902002838 Printed in the United Kingdom

Word clustering and disambiguation based on co-occurrence data[1]

HANG LI[2]

Theory NEC Laboratory, Real World Computing Partnership, c/o Internet Systems Research Laboratories, NEC Corporation, 4-1-1 Miyazaki, Miyamae-ku, Kawasaki 216-8555, Japan
e-mail: hangli@microsoft.com

(*Received 18 May 2001; revised 23 October 2001*)

Abstract

We address the problem of clustering words (or constructing a thesaurus) based on co-occurrence data, and conducting syntactic disambiguation by using the acquired word classes. We view the clustering problem as that of estimating a class-based probability distribution specifying the joint probabilities of word pairs. We propose an efficient algorithm based on the Minimum Description Length (MDL) principle for estimating such a probability model. Our clustering method is a natural extension of that proposed in Brown, Della Pietra, deSouza, Lai and Mercer (1992). We next propose a syntactic disambiguation method which combines the use of automatically constructed word classes and that of a hand-made thesaurus. The overall disambiguation accuracy achieved by our method is 88.2%, which compares favorably against the accuracies obtained by the state-of-the-art disambiguation methods.

1 Introduction

We consider the problem of clustering words, or that of constructing a thesaurus, based on co-occurrence data. We view this problem as that of estimating a class-based probability distribution specifying the joint probabilities of word pairs, such as noun verb pairs. In this paper, we assume that the distribution is expressed in the following way, which is stated for noun verb pairs for the sake of readability: The joint probability of a noun and a verb is expressed as the product of the joint probability of the noun class and the verb class which the noun and the verb respectively belong to, and the conditional probabilities of the noun and the verb given their respective classes.

As a method for estimating such a probability model, we propose an algorithm based on the Minimum Description Length (MDL) principle. Our clustering algorithm iteratively merges noun classes and verb classes in turn, in a bottom up fashion.

[1] A previous version of this paper appeared in *COLING-ACL'98* (Li and Abe 1998b).
[2] Current Address: Hang Li, Microsoft Research Asia, 5F Sigma Center, No. 49 Zhichun Road Haidian District, Beijing, China 100080. Email: hangli@microsoft.com, Home Page: http://www.research.microsoft.com/users/hangli/

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

In each iteration, it calculates the increase in data description length resulting from merging any noun (or verb) class pair, and performs the merge having the least increase in data description length, provided that the increase in data description length is less than the reduction in model description length.

There have been a number of methods proposed in the literature to address the word clustering problem (e.g. Hindle 1990; Pereira, Tishby and Lee 1993; McKeown and Hatzivassiloglou 1993; Grishman and Sterling 1994; Dagan, Pereira and Lee 1994; Grefenstette 1994; Stolcke and Omohundro 1994; Tokunaga, Iwayama and Tanaka 1995; McMahon and Smith 1996; Ushioda 1996; Hogenhout and Matsumoto 1997; Lin 1998; Rooth, Riezler, Prescher, Carroll and Beil 1999). The method proposed in this paper is a natural extension of Brown *et al.*'s method in the sense that clustering can be conducted over two different sets of words and the optimal numbers of word classes are automatically determined based on the data. Brown *et al.*'s method, which is based on Maximum Likelihood Estimation (MLE), performs a merge which would result in the least reduction in average mutual information. Our method turns out to be equivalent to performing the merge with the least reduction in mutual information, provided that the reduction is below a variable threshold.

We next propose a back-off method for resolving pp-attachment ambiguities in quadruples (e.g. 'acquire business for million'), which combines the use of automatically constructed word classes and that of a hand-made thesaurus. More specifically, for a given quadruple we first use two conditional probabilities estimated by our clustering method to make a disambiguation decision. If both of them are zero, we use conditional probabilities based on the hand-made thesaurus estimated by the tree-cut estimation method we developed (Li and Abe 1998a). Finally, if both of them are still zero, we make a decision based on which type of word, verb or noun, the preposition is most likely to attach.

Experimental results indicate that our clustering method has an advantage over Brown *et al.*'s method, namely, it has the ability of automatically determining the optimal number of word classes to be created. Experimental results also indicate that our syntactic disambiguation method is very effective. With this method, we were able to achieve the overall accuracy of 88.2% for pp-attachment disambiguation, which compares favorably against the accuracies (82.4% and 84.8%) obtained using the state-of-the-art methods (Brill and Resnik 1994; Collins and Brooks 1995).

2 Probability model

Suppose available to us are co-occurrence data over two sets of words, such as the sample of verbs and the head nouns of their direct objects given in Figure 1. Our goal is to (repeatedly) cluster the two sets of words so that words having similar co-occurrence patterns are classified in the same class, and output classes of words (or a thesaurus) for each set of words.

We can view this problem as that of estimating the best probability model from among a class of models (probability distributions) which can give rise to the co-occurrence data.

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

	eat	drink	make
wine	0	3	1
beer	0	5	1
bread	4	0	2
rice	4	0	0

Fig. 1. Example co-occurrence data.

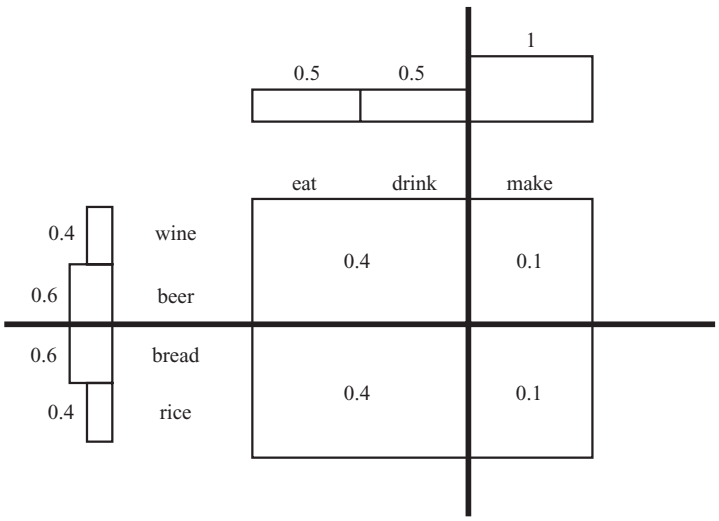


Fig. 2. Example hard clustering model.

In this paper, we consider the following type of probability model. Assume without loss of generality that the two sets of words are a set of nouns N and a set of verbs V. A partition T_n of N is a set satisfying $T_n \subseteq 2^N$, $\cup_{C_n \in T_n} C_n = N$ and $\forall C_i, C_j \in T_n$ $(i \neq j), C_i \cap C_j = \emptyset$. A partition T_v of V can be defined analogously. Each element in a partition forms a class of words. We define a *hard clustering model* with respect to a noun partition T_n and a verb partition T_v as a probability distribution of type:

$$P(n,v) = P(C_n, C_v)P(n|C_n)P(v|C_v), \quad n \in C_n, v \in C_v \tag{1}$$

where random variable n denotes a noun and random variable v a verb and where $C_n \in T_n$ and $C_v \in T_v$ are satisfied. We refer to this model as the hard clustering model, since it is based on a type of clustering in which each word can belong to only one class.

Figure 2 shows an example of the hard clustering model that can give rise to the co-occurrence data in figure 1.

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

3 Parameter estimation

A hard clustering model with a particular partition is referred to as a *discrete* hard-clustering model, in which the probability parameters are left to be estimated. The values of these parameters can be estimated based on the co-occurrence data by Maximum Likelihood Estimation. For a given set of co-occurrence data

$$S = \{(n_1, v_1), (n_2, v_2), \cdots, (n_m, v_m)\}$$

the maximum likelihood estimates of the parameters are defined as the values that maximize the following likelihood function with respect to the data:

$$\prod_{i=1}^{m} P(n_i, v_i) = \prod_{i=1}^{m} P(n_i|C_{n_i})P(v_i|C_{v_i})P(C_{n_i}, C_{v_i}), \quad n_i \in C_{n_i}, v_i \in C_{v_i}.$$

It is easy to see that this is possible by setting the parameters as

$$\hat{P}(C_n, C_v) = \frac{f(C_n, C_v)}{|S|}, \quad \hat{P}(x|C_x) = \frac{f(x)}{f(C_x)}, \quad x \in N \cup V.$$

Here, $|S|$ denotes the entire data size, $f(C_n, C_v)$ the frequency of word pairs in class pair (C_n, C_v), $f(x)$ the frequency of word x, and $f(C_x)$ the frequency of words in class C_x.

4 Model selection criterion

The question now is what criterion we should employ to select the best model from among the possible models. Here we adopt the Minimum Description Length (MDL) principle. MDL (Rissanen 1989) is a criterion for statistical estimation proposed in information theory.

In applying MDL, we calculate the code length for encoding each model, referred to as the *model description length* $L(M)$, the code length for encoding the given data through the model, referred to as the *data description length* $L(S|M)$ and their sum: $L(S : M) = L(M) + L(S|M)$. The MDL principle stipulates that for statistical estimation the best probability model with respect to given data is that which requires the least total description length. It is theoretically demonstrated that as a criterion for statistical estimation MDL has many good properties (Rissanen 1989; Barron and Cover 1991; Yamanishi 1992).[3] Particularly, the optimal number of parameters can be estimated by using MDL.

The data description length is calculated as

$$L(S|M) = -\sum_{(n,v)\in S} \log P_{\hat{\theta}}(n, v),$$

where $\hat{\theta}$ stands for the maximum likelihood estimates of the parameters defined in Section 3.

[3] For an introduction to MDL, see for example, Li (1998).

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

The model description length is calculated as

$$L(M) = \frac{k}{2} \log |S|,$$

where k denotes the number of *free* parameters in the model, and $|S|$ the entire data size. In this paper, we ignore the code length for encoding a discrete model, assuming that they are equal for all models and consider only the code length for encoding the parameters of a model as the model description length.

We note that there are alternative ways for calculating $L(S : M)$. For example, we can calculate it as

$$L(S : M) = -\sum_{(n,v)\in S} \log P_{\hat{\theta}}(n, v) + \frac{k}{2} \log \frac{|S|}{2\pi} + \log \frac{\pi^{(k+1)/2}}{\Gamma\left(\frac{k+1}{2}\right)},$$

where Γ denotes the Gamma function and π the circular constant. Since these alternatives are approximations of one another asymptotically, here we use only the simplest formulation.

If computation time were of no concern, we could in principle calculate the total description length for each model and select the optimal model in terms of MDL. Since the number of hard clustering models is of order $O(|N|^{|N|}|V|^{|V|})$ (Li 1998), where $|N|$ and $|V|$ denote the size of the noun set and the verb set, respectively, it would be infeasible to do so. We therefore need to devise an efficient algorithm that heuristically performs this task.

5 Algorithm

The proposed algorithm, which we call '2D-Clustering', iteratively selects an optimal MDL-model from among those hard clustering models which can be obtained from the current model by merging a noun (or verb) class pair. As it turns out, the minimum description length criterion can be re-formalized in terms of average mutual information, and the algorithm can be formulated to calculate, in each iteration, the reduction of mutual information which would result from merging any noun (or verb) class pair, and perform the merge having the least mutual information reduction, provided that the reduction is below a variable threshold.

2D-Clustering (S, b_n, b_v)
(S is the input co-occurrence data, and b_n and b_v are positive integers.)

1. Initialize the set of noun classes T_n and the set of verb classes T_v as: $T_n = \{\{n\}|n \in N\}, T_v = \{\{v\}|v \in V\}$ where N and V denote the noun set and the verb set, respectively.
2. Repeat the following three sub-steps:
   a) Execute Merge (S, T_n, T_v) b_n times to update T_n,
   b) Execute Merge (S, T_v, T_n) b_v times to update T_v,
   c) If T_n and T_v are unchanged, go to Step 3.
3. Output T_n and T_v (or construct and output a thesaurus for nouns based on the history of T_n, and one for verbs based on the history of T_v).

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

Next, we describe the algorithm of Merge, as it is being applied to the set of noun classes T_n with the set of verb classes T_v fixed.

Merge (S, T_n, T_v)

1. For each class pair in T_n, calculate the reduction of mutual information (2) which would result from merging them. (The details will follow.) Discard those class pairs whose mutual information reduction is not less than the threshold (3). Find the class pair with the least mutual information reduction.
2. Merge the class pair with the least mutual information reduction.
3. Output current T_n.

Next, we explain why the criterion in terms of description length can be reformalized in terms of mutual information. Let M_B denote the model before a merge and M_A the model after the merge. Let ΔL_{dat} denote the increase in data description length

$$\Delta L_{\mathrm{dat}} = L(S|M_A) - L(S|M_B) > 0.$$

In addition, suppose that M_A is obtained by merging two noun classes C_i and C_j in M_B to a single noun class C_{ij}. We need only calculate the difference between data description lengths with respect to these classes.

$$\begin{aligned}\Delta L_{\mathrm{dat}} &= -\sum_{C_v\in T_v}\left(\sum_{n\in C_{ij}}\sum_{v\in C_v}\log P_{\hat{\theta}}(n,v)\right)+\sum_{C_v\in T_v}\left(\sum_{n\in C_i}\sum_{v\in C_v}\log P_{\hat{\theta}}(n,v)\right)\\ &\quad+\sum_{C_v\in T_v}\left(\sum_{n\in C_j}\sum_{v\in C_v}\log P_{\hat{\theta}}(n,v)\right).\end{aligned}$$

Since from (1)

$$P(n,v) = \frac{P(C_n,C_v)}{P(C_n)P(C_v)}P(n)P(v)$$

holds, we have

$$\begin{aligned}\frac{1}{|S|}\Delta L_{\mathrm{dat}} &= \frac{1}{|S|}\left(-\sum_{C_v\in T_v} f(C_{ij},C_v)\log\frac{\hat{P}(C_{ij},C_v)}{\hat{P}(C_{ij})\hat{P}(C_v)}\right.\\ &\quad+\sum_{C_v\in T_v} f(C_i,C_v)\log\frac{\hat{P}(C_i,C_v)}{\hat{P}(C_i)\hat{P}(C_v)}\\ &\quad\left.+\sum_{C_v\in T_v} f(C_j,C_v)\log\frac{\hat{P}(C_j,C_v)}{\hat{P}(C_j)\hat{P}(C_v)}\right).\end{aligned}$$

This is a quantity equivalent to the mutual information reduction. Note that average mutual information between T_n and T_v is defined as

$$I(T_n,T_v) = \sum_{C_n\in T_n}\sum_{C_v\in T_v} P(C_n,C_v)\log\frac{P(C_n,C_v)}{P(C_n)P(C_v)}.$$

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

We further have

$$\frac{1}{|S|}\Delta L_{\text{dat}} = \frac{1}{|S|}\left(-\sum_{C_v \in T_v}(f(C_i,C_v)+f(C_j,C_v))\log\frac{f(C_i,C_v)+f(C_j,C_v)}{f(C_i)+f(C_j)}\right.$$

$$\left.+\sum_{C_v \in T_v} f(C_i,C_v)\log\frac{f(C_i,C_v)}{f(C_i)}+\sum_{C_v \in T_v} f(C_j,C_v)\log\frac{f(C_j,C_v)}{f(C_j)}\right). \quad (2)$$

Let ΔL_{mod} denote the reduction in model description length

$$\Delta L_{\text{mod}} = L(M_B) - L(M_A) > 0.$$

Note that ΔL_{mod} is equivalent for each merge. Specifically, we have

$$\frac{1}{|S|}\Delta L_{\text{mod}} = \frac{(|T_v|-1)\log|S|}{2|S|}. \quad (3)$$

According to MDL, a model M_A with the least ΔL_{dat} and satisfying

$$\Delta L_{\text{dat}} < \Delta L_{\text{mod}}$$

should be selected. Equivalently, a model M_A with the least mutual information reduction in (2) should be selected, provided that the reduction is smaller than the threshold in (3).

6 Efficient implementation

In our implementation of the algorithm, we first load the co-occurrence data into a matrix, with nouns (or noun classes) corresponding to rows, verbs (or verb classes) to columns.

For each row i, we maintain

$$s(i) = \sum_{C_v \in T_v} f(C_i,C_v)\log\frac{f(C_i,C_v)}{f(C_i)}. \quad (4)$$

For each row pair i and j ($i < j$), we maintain

$$d(i,j) = \sum_{C_v \in T_v}(f(C_i,C_v)+f(C_j,C_v))\log\frac{f(C_i,C_v)+f(C_j,C_v)}{f(C_i)+f(C_j)}. \quad (5)$$

For each column l,

$$s(l) = \sum_{C_n \in T_n} f(C_n,C_l)\log\frac{f(C_n,C_l)}{f(C_l)}.$$

For each column pair l and m ($l < m$),

$$d(l,m) = \sum_{C_n \in T_n}(f(C_n,C_l)+f(C_n,C_m))\log\frac{f(C_n,C_l)+f(C_n,C_m)}{f(C_l)+f(C_m)}.$$

We then execute Merge by using the quantities. Specifically, we can calculate the increase in data description length that would result from merging row i and j as

$$\Delta L_{\text{dat}} = -d(i,j) + s(i) + s(j)$$

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

and calculate the increase in data description length that would result from merging column l and m as

$$\Delta L_{\mathrm{dat}} = -d(l,m) + s(l) + s(m).$$

Suppose that we are to merge the row i' and j' ($i' < j'$). Then, for each column l we rewrite $s(l)$ as

$$\begin{aligned} s(l) \quad := \quad & s(l) - \Bigg(f(C_{i'}, C_l) \log \frac{f(C_{i'}, C_l)}{f(C_l)} + f(C_{j'}, C_l) \log \frac{f(C_{j'}, C_l)}{f(C_l)} \\ & - (f(C_{i'}, C_l) + f(C_{j'}, C_l)) \log \frac{f(C_{i'}, C_l) + f(C_{j'}, C_l)}{f(C_l)} \Bigg). \end{aligned}$$

For each column pair l and m we rewrite $d(l,m)$ as

$$\begin{aligned} d(l,m) \quad := \quad & d(l,m) - \Bigg((f(C_{i'}, C_l) + f(C_{i'}, C_m)) \log \frac{f(C_{i'}, C_l) + f(C_{i'}, C_m)}{f(C_l) + f(C_m)} \\ & + (f(C_{j'}, C_l) + f(C_{j'}, C_m)) \log \frac{f(C_{j'}, C_l) + f(C_{j'}, C_m)}{f(C_l) + f(C_m)} \\ & - (f(C_{i'}, C_l) + f(C_{j'}, C_l) + f(C_{i'}, C_m) + f(C_{j'}, C_m)) \\ & \times \log \frac{f(C_{i'}, C_l) + f(C_{j'}, C_l) + f(C_{i'}, C_m) + f(C_{j'}, C_m)}{f(C_l) + f(C_m)} \Bigg). \end{aligned}$$

For each column l, we add $f(C_{i'}, C_l)$ and $f(C_{j'}, C_l)$ and write it on row i', move $f(C_{\mathrm{last}}, C_l)$ to row j', and reduce the matrix by one row. For the new row i', we recalculate $s(i')$ as in (4). For each new-row pair between i' and i ($i \neq i'$), we recalculate $d(i', i)$ when $i' < i$, or $d(i, i')$ when $i < i'$ as in (5).

With the above implementation, the time complexity of the algorithm is only of order $O(|N|^3 + |V|^3)$ where $|N|$ denotes the size of the noun set, $|V|$ that of the verb set.

7 Related work

7.1 *Brown* et al. *(1992)*

The hard clustering model defined in (1) can be considered to be an extension of the model proposed by Brown *et al.* First, dividing (1) by $P(v)$, we obtain

$$\frac{P(n,v)}{P(v)} = P(C_n|C_v)P(n|C_n)\frac{P(C_v)P(v|C_v)}{P(v)}.$$

Since hard clustering implies

$$\frac{P(C_v)P(v|C_v)}{P(v)} = 1$$

holds, we have

$$P(n|v) = P(C_n|C_v)P(n|C_n). \tag{6}$$

In this way, the hard clustering model turns out to be a class-based bigram model and is similar to Brown *et al.*'s model. The difference is that the model of (6) assumes

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

that the clustering for C_n and the clustering for C_v can be different, while the model of Brown *et al.* assumes that they are the same.

Brown *et al.*'s method is based on Maximum Likelihood Estimation (MLE), but it only employs MLE *locally*. In general, MLE is not able to select the best model from a class of models having different numbers of parameters because MLE will always suggest selecting the model having the largest number of parameters, which would have a better fit to the given data. In Brown *et al.*'s case, MLE is used to iteratively select the model with the maximum likelihood from a class of models that have the same number of parameters. Such a model class is repeatedly obtained by merging any word class pair in the current situation. The number of word classes within the models in the final model class, therefore, has to be designated in advance. There is, however, no guarantee at all the designated number will be optimal.

The method proposed here resolves this problem by employing MDL. This is reflected in use of the threshold (3) in clustering, which will result in automatic selection of the optimal number of word classes to be created.

7.2 Li and Abe (1996)

We can restrict the hard clustering model (1) by assuming that words within a same class are generated with an equal probability, obtaining

$$P(n,v) = P(C_n, C_v)\frac{1}{|C_n|}\frac{1}{|C_v|}$$

which is equivalent to the model proposed by Li and Abe (1996). Employing this restricted model has the undesirable tendency to classify into different classes those words that have similar co-occurrence patterns but have different absolute frequencies.

7.3 Pereira et al. *(1993)*

A very general model of noun-verb co-occurrence is a model of the following form:

$$P(n,v) = \sum_{C_n \in \Gamma_n} \sum_{C_v \in \Gamma_v} P(C_n, C_v)P(n|C_n)P(v|C_v). \tag{7}$$

Here Γ_n denotes a set of not necessarily disjoint noun classes satisfying

$$\sum_{C_n \in \Gamma_n} C_n = N.$$

Similarly Γ_v denotes a set of verb classes. We can view the problem of clustering words in general as that of estimating such a model. This type of clustering in which a word can belong to several different classes is generally referred to as *soft clustering*. If we assume in the above model that each verb forms a verb class by itself, then (7) becomes

$$P(n,v) = \sum_{C_n \in F_n} P(C_n, v)P(n|C_n)$$

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

which is equivalent to the model in Pereira *et al.* (1993). On the other hand, if we restrict the general model of (7) so that both noun classes and verb classes are disjoint, then we obtain the hard clustering model we propose here (1). All of these models, therefore, are some special cases of (7). Each specialization comes with its merit and demerit. For example, employing a model of soft clustering will make the clustering process more flexible but also make the learning process more computationally demanding. Hard clustering obviously has the merits and demerits of soft clustering reversed.

8 Disambiguation method

We next make use of the acquired word classes, more specifically the obtained hard clustering models, in syntactic disambiguation. In particular, we consider the problem of resolving pp-attachment ambiguities in quadruples like 'acquire business for million' (e.g. Hindle and Rooth 1991; Sekine, Carroll, Ananiadou, and Tsujii 1992; Resnik 1993; Alshawi and Carter 1994; Ratnaparkhi, Reynar and Roukos 1994; Brill and Resnik 1994; Collins and Brooks 1995; Abe and Li 1996; Zavrel, Daelemans and Veenstra 1997; Ratnaparkhi 1998; Stetina and Nagao 1998). We determine to which of 'acquire' or 'business' the phrase 'for million' should be attached.

Let us denote a quadruple as (v, n_1, p, n_2). We can perform pp-attachment disambiguation by comparing the probabilities $P_p(n_2|v)$ and $P_p(n_2|n_1)$. If the former is larger, we attach (p, n_2) to v; if the latter is larger we attach it to n_1; otherwise we make no decision.

Since the number of probabilities to be estimated is extremely large, estimating all of these probabilities accurately is generally infeasible (i.e. the data sparseness problem). Using our clustering model to calculate these conditional probabilities can solve this problem.

We in fact propose a back-off method for the disambiguation purpose:

1. Receive triples (v, p, n_2) and (n_1, p, n_2) as training data.
2. Calculate the two conditional probabilities $P_p(n_2|v)$ and $P_p(n_2|n_1)$ using hard clustering models constructed by our method and make a disambiguation decision.
3. If both of the probabilities are zero, calculate them with a hand-made thesaurus using the 'tree cut' estimation method we developed (Li and Abe 1998a).[4] Make a disambiguation decision with the probabilities.
4. If both of the probabilities are still zero, calculate the conditional probabilities $P(1|p)$ and $P(0|p)$ from triples where 1 denotes the attachment to v and 0 that to n. Make a disambiguation decision with the probabilities.
5. Finally, if both of them are still zero, attach (p, n_2) to n_1 as a default decision.

[4] The method outputs a tree cut in a given thesaurus with conditional probabilities attached to all the nodes (word classes) in the tree cut. MDL is used to select the best tree cut model.

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

In section 9, we will make a justification of this disambiguation method based on experimental results.

9 Experimental results

9.1 Qualitative evaluation

In this experiment, we used heuristic rules to extract verbs and the head nouns of their direct objects from the tagged WSJ corpus (ACL/DCI CD-ROM1) consisting of 126,084 sentences.

We then conducted word clustering on these data using our method. Figure 3 shows a clustering result for 400 randomly selected nouns, based on their appearances as direct objects of 40 randomly selected verbs (some words are omitted due to space limitation). The clustering result seems to agree with human intuition to some degree, although it is constructed based on a relatively small amount of co-occurrence data.

9.2 Evaluation with perplexity

We randomly selected from the tagged WSJ corpus 1,000 nouns that had more than five occurrences, and extracted 8925 compound noun doubles containing those nouns (e.g. 'air control') from the same corpus. We randomly generated five data sets consisting of different training and test data for five-fold cross validation experiments.

We performed clustering on the training data in each data set by using both our method (2D-Clustering) and Brown *et al.*'s method (Brown). We actually implemented an extended version of their method, which separately conducts clustering for nouns on the left and those on the right (which should only improve the performance).

We next calculated the perplexities for predicating the test data in each data set using the probabilities estimated by 2D-Clustering and Brown. (The test data perplexity is a measure of testing how well an estimated probability model predicts future data, and is defined as

$$2^{H(P_T,P_M)}, H(P_T,P_M) = -\sum_x P_T(x)\log P_M(x),$$

where $P_M(x)$ denotes the estimated model, $P_T(x)$ the empirical distribution of the test data (cf. Bahl, Jelinek and Mercer 1983). It is roughly the case that the smaller perplexity a model has, the closer to the true model it is.) We also tested the method of using the probabilities estimated based on word co-occurrences, denoted as 'Word-based'. Figure 4 shows the perplexity values and the numbers of the obtained word classes for the three methods, averaged over five trials. Since for Brown the number of classes finally created has to be decided in advance, we tried a number of alternatives and obtained results for each of them. (Note that for 2D-Clustering the optimal number of classes is automatically selected.)

We see that the perplexity of 2D-Clustering approaches that of the minimum of

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

Verb classes:
0: acquire hold buy purchase sell
1: add start close open create run end
2: avoid stop cover require put help protect support show see build
3: boost increase cut reduce pay raise

Noun classes:
0: share asset
1: stake
2: stock insurance chain corp. co. mortgage estate ltd. Inc. inc
3: company business operation
4: rate tax wage size salary fine estimate fare exposure dividend
5: price cost level export circulation discount rating spending earning
6: oct. sept. aug. july jan. Saturday dec. march Tuesday Monday june
7: year firm today bank group hundred
8: interest concern executive note package portion rest equity
9: office Friday Wednesday store april post studio branch session
10: yesterday cent point campaign week trust nov.
11: order carrier u.s. child idea material protection assistance
12: money charge profit emphasis financing inflation effect flag
13: debt possibility
14: information freedom housing egg strategy increase technology
15: service image term power way rule mind proposal people client
16: risk capacity barrier provision volatility
17: product computer record equipment piece park maker house car machine
18: sale supply government base bid margin yield fear subsidy chance
19: revenue premium attention fee expens output deficit limit demand
20: role hand result increase proceed return figure efficiency
21: unit subsidiary columbia corp division talk
22: fund deal mark activity offering position arm line time million
23: land thrift item hearing conference meeting title future magazine
24: system project network state game status software vehicle policy
25: month book venture room world ad course
26: support country measure suit evidence cut plan statement end source
27: market industry life account day job show surplus voice problem
28: amount shareholder inventory holding
29: number bill staff income competition force cash volume capital
30: approval family buyer decline improvement help prize clearance
31: loan kind issue total farmer dollar portfolio aircraft majority oil
32: tie worker strength need standard practice loss aid credit benefit

Fig. 3. Example clustering result.

Brown, indicating that 2D-Clustering has the ability of automatically selecting the optimal number of word classes.

9.3 PP-attachment disambiguation

In terms of data we used the bracketed WSJ corpus (Penn Tree Bank 1) (Marcus, Santorini and Marcinkiewicz 1993). We first randomly selected one of the 26 file

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

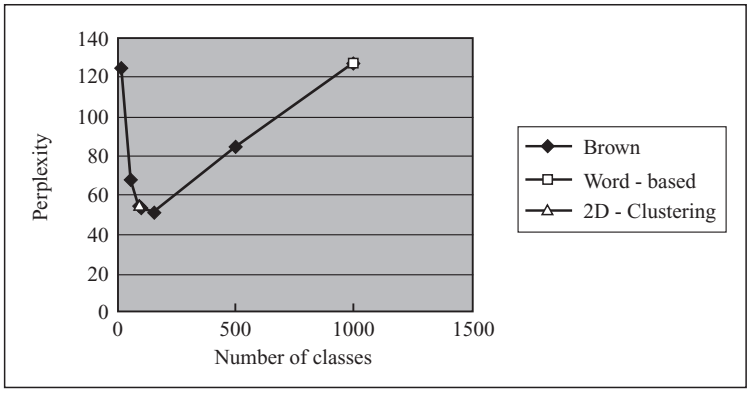


Fig. 4. Evaluation results.

Table 1. *Number of data items*

	Training data
Average number of triples per data set	91218.1
Average number of triples related to test data	11304.9
Average number of quadruples per data set	21656.6
Test data	
Average number of quadruples per data set	820.4

directories of the WSJ corpus as test data and used what remained as training data. We repeated this process ten times and obtained ten data sets consisting of different training and test data. We used these ten data sets to conduct cross validation. That is, all of the experimental results reported below were obtained by taking an average over ten trials. From the test data in each data set, we extracted (v, n_1, p, n_2) quadruples (e.g. 'acquire business for million') using a tool provided by the Tree Bank called 'tgrep'. At the same time, we obtained the *answer* for the pp-attachment for each quadruple. From the training data of each data set, we extracted (v, p, n_2) and (n_1, p, n_2) triples (e.g. 'acquire for billion' and 'business for person') using a tool we developed. We also extracted quadruples from the training data as before. (A more challenging task is to take an unsupervised approach to disambiguation (e.g. Ratnaparkhi 1998; Pantel and Lin 2000).

Table 1 shows the number of different types of data obtained in the above process. Here 'A triple related to test data' is a kind of triple in which the head word and preposition are either in common with (v, p) or with (n_1, p) in a quadruple (v, n_1, p, n_2) in test data.

We performed clustering on triple data using our method (2D-Clustering) and Brown *et al.*'s method (Brown). For both methods, following the proposal due to Tokunaga, Iwayama and Tanaka (1995), we separately conducted clustering with respect to each of the 9 most frequently occurring prepositions ('of', 'for', 'with', etc.). We did not cluster words for rarely occurring prepositions. We then performed disambiguation based on 2D-Clustering and Brown. We also tested the method of

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

Table 2. *PP-attachment disambiguation results*

Method	Coverage (%)	Accuracy (%)
Word-based	30.5	95.4
Brown (best)	50.9	98.6
2D-Clustering	50.8	98.6
WordNet	74.3	94.5
Auto-Thesaurus	42.6	97.1

Table 3. *PP-attachment disambiguation results with 100% coverage*

Method	Accuracy (%)
Default	56.2
Word-based + Default	68.3
Brown + Default	76.2
2D-Clustering + Default	76.2
WordNet + Default	83.7
Auto-Thesaurus + Default	73.8

using the conditional probabilities estimated based on word co-occurrences, denoted as 'Word-based'.

Next, rather than using the conditional probabilities estimated by our method, we only used the noun thesauruses constructed by our method, and applied the method we previously developed (Li and Abe 1998a) to estimate the best tree cut models within the thesauruses in order to estimate the conditional probabilities $P_p(n_2|v)$ and $P_p(n_2|n_1)$. We call the disambiguation method 'Auto-Thesaurus'. Finally, we tried the disambiguation method we proposed in Li and Abe (1998a), which uses a hand-made thesaurus, WordNet.

Table 2 shows the results for all these methods in terms of coverage and accuracy. Here *coverage* refers to the percentage of test data by which a disambiguation method can reach a decision, and *accuracy* refers to the proportion of correct decisions among all decisions made.

To obtain 100% coverage, we enhanced each of these methods by using a default rule (to always attach (p, n_2) to n_1 when a decision cannot be made, which is indicated as '+Default'. Table 3 shows the results of these experiments.

We can make a number of observations from these results. 2D-Clustering achieves a broader coverage than Auto-Thesaurus. This is because in order to estimate the probabilities for disambiguation, the former exploits more information than the latter (two dimensional class information versus one dimensional class information).

For Brown, we show here only its best result, which happens to be the same as the result for 2D-Clustering, but in order to obtain this result we had to take the trouble of conducting a number of tests to find the best level of clustering. For 2D-Clustering, this was done once and automatically. Therefore we conclude that our method improves this previous method. Figure 5 shows different results of Brown with different settings (i.e. different designated numbers of word classes), and those of 2D-Clustering and Word-based. We found that in many cases both

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

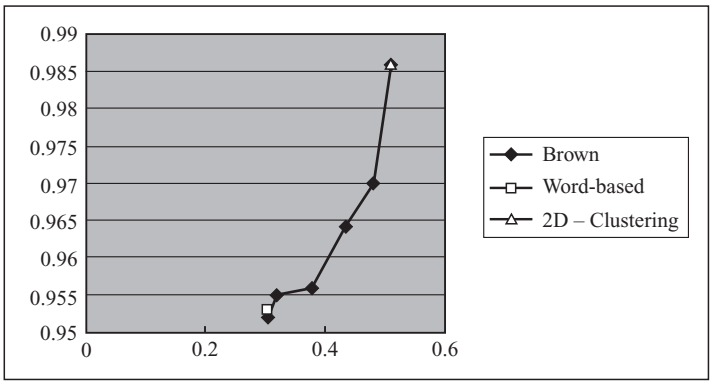


Fig. 5. PP-attachment disambiguation results.

Table 4. *Comparison with previous methods*

Method	Accuracy (%)
Back-off	88.2
Brill-Resnik	82.4
Collins-Brooks	84.8

the probabilities $P_p(n_2|v)$ and $P_p(n_2|n_1)$ were estimated as zero by Word-based. By clustering words with 2D-Clustering or Brown one of the probabilities would become larger than zero and a disambiguation decision could be made. The accuracy of such disambiguation decisions was generally very high and this was the main reason for the increases of both coverage and accuracy for 2D-Clustering and Brown.

2D-Clustering outperforms WordNet in terms of accuracy, but not in terms of coverage. This seems reasonable, since automatically constructed word classes are more domain dependent and therefore capture the domain dependent features better, and thus can help achieve higher accuracy. On the other hand, with the relatively small size of training data we had available, its coverage is smaller than that of a general purpose hand made thesaurus. The result indicates that it makes sense to combine automatically constructed word classes and a hand-made thesaurus, as we have proposed in section 8.

We next tested the method we proposed in Section 8, denoted here as 'Back-off'. We also tested the state-of-the-art methods for pp-attachment disambiguation proposed in Brill and Resnik (1994) and Collins and Brooks (1995) using the same data sets. We denote them as Brill-Resnik and Collins-Brooks, respectively. Table 4 shows the results, and figure 6 gives the accuracy of each method for each of the ten trials. We see that Back-off always outperforms Brill-Resnik and Collins-Brooks.

Stetina and Nagao (1998) reported an accuracy of 88.1% in a pp-attachment disambiguation experiment using their method. The data they used was the same as ours, but the splitting of the data to training and test was different. And thus their results can not be directly compared with ours. (We did not implement their method in our experiment.)

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

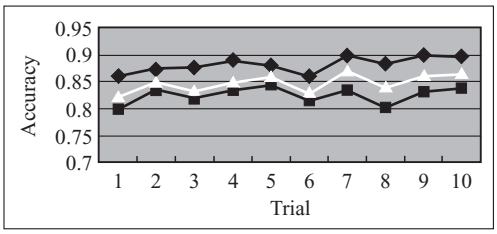


Fig. 6. Cross-validation results.

We note that Brill-Resnik and Collins-Brooks require quadruples and *disambiguation decisions* as training data, while Back-off uses only triples as training data which could be extracted from sentences without ambiguities (cf. "A girl with telescope saw me"). Thus there would be in general more training data available for Back-off in practice.

We think that the high performance of Back-off is due to two facts: (1) it is a class-based approach to disambiguation and thus is more robust to the data sparseness problem than the two previous methods, which are word-based; (2) there are more triples available as training data than quadruples in the corpus data.

10 Conclusions

We have proposed a new method of clustering words based on co-occurrence data. Our method employs a probability model which naturally represents co-occurrence patterns over word pairs, and makes use of an efficient estimation algorithm based on the MDL principle. Our clustering method improves upon the previous method proposed by Brown *et al.*, and furthermore it can be used to derive a disambiguation method with overall disambiguation accuracy of 88.2%, which improves the performances of state-of-the-art disambiguation methods.

Acknowledgements

I am very grateful to Naoki Abe and Kenji Yamanishi for their encouragement and advice.

References

Alshawi, H. and Carter, D. (1994) Training and scaling preference functions for disambiguation. *Computational Linguistics* **20**(4): 635–648.

Abe, N. and Li, H. (1996) Learning word association norms using tree cut pair models. *Proceedings 13th International Conference on Machine Learning*, pp. 3–11.

Barron, A. R. and Cover, T. M. (1991) Minimum complexity density estimation. *IEEE Transaction on Information Theory* **37**(4): 1034–1054.

Bahl, L. R., Jelinek, F. and Mercer, R. (1983) A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **5**(2): 179–190.

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C. and Mercer, R. L. (1992) Class-based n-gram models of natural language. *Computational Linguistics* **18**(4): 283–298.

Brill, E. and Resnik, P. (1994) A rule-based approach to prepositional phrase attachment disambiguation. *Proceedings 15th International Conference on Computational Linguistics*, pp. 1198–1204.

Collins, M. and Brooks, J. (1995) Prepositional phrase attachment through a backed-off model. *Proceedings 3rd Workshop on Very Large Corpora.*

Dagan, I., Pereira, F. and Lee, L. (1994) Similarity-based estimation of word cooccurrence probabilities. *Proceedings 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 272–278.

Grefenstette, G. (1994) *Explorations in Automatic Thesaurus Discovery*. Kluwer.

Grishman, R. and Sterling, J. (1994) Generalizing automatically generated selectional patterns. *Proceedings 15th International Conference on Computational Linguistics*, pp. 742–747.

Hindle, D. (1990) Noun classification from predicate-argument structures. *Proceedings 28th Annual Meeting of the Association for Computational Linguistics*, pp. 268–275.

Lin, D. (1998) Automatic retrieval and clustering of similar words. *Proceedings 18th International Conference on Computational Linguistics and the 36th Annual Meeting of Association for Computational Linguistics*, pp. 768–774.

Hogenhout, W. R. and Matsumoto, Y. (1997) A preliminary study of word clustering based on syntactic behavior. *CoNLL97: Computational Natural Language Learning, Proceedings 1997 Meeting of the ACL Special Interest Group in Natural Language Learning*, pp. 16–24.

Hindle, D. and Rooth, M. (1991) Structural ambiguity and lexical relations. *Proceedings 29th Annual Meeting of the Association for Computational Linguistics*, pp. 229–236.

Li, H. (1998) A Probabilistic Approach to Lexical Semantic Knowledge Acquisition and Structural Disambiguation. PhD Thesis, University of Tokyo.

Li, H. and Abe, N. (1996) Clustering words with the MDL principle. *Proceedings 16th International Conference on Computational Linguistics*, pp. 4–9.

Li, H. and Abe, N. (1998a) Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics* **24**(2): 217–244.

Li, H. and Abe, N. (1998b) Word clustering and disambiguation based on co-occurrence data. *Proceedings 18th International Conference on Computational Linguistics and 36th Annual Meeting of Association for Computational Linguistics*, pp. 749–755.

McKeown, K. and Hatzivassiloglou, V. (1998) Augmenting lexicons automatically: Clustering semantically related adjectives. *Proceedings ARPA Workshop on Human Language Technology*.

McMahon, J. C. and Smith, F. J. (1996) Improving statistical language model performance with automatically generated word hierarchies. *Computational Linguistics* **22**(2): 217–247.

Marcus, M. P., Santorini, B. and Marcinkiewicz, M. A. (1993) Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* **19**(1): 313–330.

Pantel, P. and Lin, D. (2000) An unsupervised approach to prepositional phrase attachment using contextually similar words. *Proceedings 38th Annual Meeting of the Association for Computational Linguistics*, pp. 101–108.

Pereira, F., Tishby, N. and Lee, L. (1993) Distributional clustering of English words. *Proceedings 31st Annual Meeting of the Association for Computational Linguistics*, pp. 183–190.

Ratnaparkhi, A. (1998) Statistical models for unsupervised prepositional phrase attachment. *Proceedings 36th Annual Meeting of Association for Computational Linguistics and 17th International Conference on Computational Linguistics,* pp. 1079–1085.

Resnik, P. (1993) Selection and Information: A Class-based Approach to Lexical Relationships. PhD Thesis, University of Pennsylvania.

Rissanen, J. (1989) *Stochastic Complexity in Statistical Inquiry*. World Scientific.

Rooth, M., Riezler, S., Prescher, D., Carroll, G. and Beil, F. (1999) Inducing a semantically annotated lexicon via EM-based clustering. *Proceedings 36th Annual Meeting of the Association for Computational Linguistics.*

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press

Ratnaparkhi, A., Reynar, J. and Roukos, S. (1994) A maximum entropy model for prepositional phrase attachment. *Proceedings ARPA Workshop on Human Language Technology*, pp. 250–255.

Sekine, S., Carroll, J. J., Ananiadou, S. and Tsujii, J. (1992) Automatic learning for semantic collocation. *Proceedings 3rd Conference on Applied Natural Language Processing*, pp. 104–110.

Stetina, J. and Nagao, M. (1998) PP attachment ambiguity resolution through supervised learning. *J. Natural Language Processing* **5**(1): 37–57.

Stolcke, A. and Omohundro, S. (1994) Inducing probabilistic grammars by Bayesian model merging. In: Carrasco, R. C. and Oncina, J. (eds.), *Grammatical Inference and Applications*, pp. 106–118. Springer-Verlag.

Tokunaga, T., Iwayama, M. and Tanaka, H. (1995) Automatic thesaurus construction based-on grammatical relations. *Proceedings 14th International Joint Conference on Artificial Intelligence*, pp. 1308–1313.

Ushioda, A. (1996) Hierarchical clustering of words and application to NLP tasks. *Proceedings 4th Workshop on Very Large Corpora,* pp. 28–41.

Yamanishi, K. (1992) A learning criterion for stochastic rules. *Machine Learning* **9**: 165–203.

Zavrel, J., Daelemans, W. and Veenstra, J. (1997) Resolving PP attachment ambiguities with memory-based learning. *CoNLL97: Computational Natural Language Learning, Proceedings 1997 Meeting of the ACL Special Interest Group in Natural Language Learning*, pp. 136–144.

https://doi.org/10.1017/S1351324902002838 Published online by Cambridge University Press