Semi-supervised Clustering for Word Instances and Its Effect on Word Sense Disambiguation

Kazunari Sugiyama and Manabu Okumura

Precision and Intelligence Laboratory, Tokyo Institute of Technology,
4259 Nagatsuta, Midori, Yokohama, Kanagawa 226-8503, Japan
sugiyama@lr.pi.titech.ac.jp, oku@pi.titech.ac.jp

Abstract. We propose a supervised word sense disambiguation (WSD) system that uses features obtained from clustering results of word instances. Our approach is novel in that we employ semi-supervised clustering that controls the fluctuation of the centroid of a cluster, and we select seed instances by considering the frequency distribution of word senses and exclude outliers when we introduce "must-link" constraints between seed instances. In addition, we improve the supervised WSD accuracy by using features computed from word instances in clusters generated by the semi-supervised clustering. Experimental results show that these features are effective in improving WSD accuracy.

1 Introduction

Many words have multiple meanings depending on the context in which they are used. For example, among the possible senses of the verb "run" are "to move fast by using one's feet" and "to direct or control." Word sense disambiguation (WSD) is the task of determining the meaning of such an ambiguous word in its context. In this paper, we apply semi-supervised clustering by introducing sense-tagged instances (we refer to them as "seed instances" in the following) to the supervised WSD process. Our approach is based on the following intuitions: (1) in the case of word instances, we can use sense-tagged word instances from various sources as supervised instances, and (2) the features computed from word instances in clusters generated by our semi-supervised clustering are effective in supervised WSD since word instances clustered around sense-tagged instances may have the same sense. Existing semi-supervised clustering approaches solely focus on introducing constraints and learning distances and overlook control of the fluctuation of the cluster's centroid. In addition, to enable highly accurate semi-supervised clustering, it is important to consider how to select seed instances and how to introduce constraints between the seed instances. Regarding seed instances, we have to pay attention to the frequency distribution of word senses when selecting seed instances as well as the way of introducing "must-link" constraints, since outlier instances may exist when we select seed instances with the same sense.

In this paper, we describe our semi-supervised clustering approach that controls the fluctuation of the centroid of a cluster and propose a way of introducing appropriate seed instances and constraints. In addition, we explain our WSD approach using features computed from word instances that belong to clusters generated by the semi-supervised

A. Gelbukh (Ed.): CICLing 2009, LNCS 5449, pp. 266–279, 2009.

clustering. Our approach is novel in that we employ semi-supervised clustering that controls the fluctuation of the centroid of a cluster, and we select seed instances by considering the frequency distribution of word senses and exclude outliers when we introduce "must-link" constraints between seed instances.

2 Related Work

2.1 Semi-supervised Clustering

The semi-supervised clustering methods can be classified into *constraint-based* and *distance-based*. Constraint-based methods rely on user-provided labels or constraints to guide the algorithm toward a more appropriate data partitioning. For example, Wagstaff et al. [12,13] introduced two types of constraint – "must-link" (two instances have to be together in the same cluster) and "cannot-link" (two instances have to be in different clusters) – and their semi-supervised K-means algorithm generates data partitions by ensuring that none of the user-specified constraints are violated. Basu et al. [18] also developed a semi-supervised K-means algorithm that makes use of labeled data to generate initial seed clusters and to guide the clustering process. In distance-based approaches, an existing clustering algorithm that uses a particular clustering measure is employed; however, it is trained to satisfy the labels or constraints in the supervised data [5,9,1].

2.2 Word Sense Disambiguation

In order to improve WSD accuracy, several works add features to original features such as POS tags, local collocations, bag-of-words, syntactic relations. For example, Agirre et al. [7] proposed the idea of "topic signatures." They first submit synonyms, gloss, hypernyms, hyponyms, meronums, holonyms and attributes in WordNet as well as the target word as a query to a search engine and then compute χ^2 values (topic signatures) using the extracted words from the searched documents. Finally, they apply these topic signatures to WSD. Specia et al. [19] presented a WSD system employing inductive logic programming [16] that can represent substantial knowledge to overcome the problem of relying on a limited knowledge representation and generate a disambiguation model by applying machine learning algorithms to attribute-value vectors. Cai et al. [3] constructed topic features on an unlabeled corpus by using the latent dirichlet allocation (LDA) algorithm [6], then used the resulting topic model to tag the bag-of-words in the labeled corpus with topic distributions. Finally, to create the supervised WSD system, they constructed a classifier applying features such as POS tags, local collocations, bag-of-words, syntactic relations as well as topic models to a support vector machine.

Generally, in the case of using context as features for WSD, the feature space tends to be sparse. Niu et al. [23] proposed a semi-supervised feature clustering algorithm to conduct dimensionality reduction for WSD with maintaining its accuracy.

Other recent WSD studies include nominal relationship classification where pattern clusters are used as the source of machine learning features to learn a model [4], and WSD system using OntoNotes project [8] that has a coarse-grained sense inventory [24].

3 Proposed Method

The existing semi-supervised clustering approaches solely focus on introducing constraints and learning distances. However, when we apply semi-supervised clustering to word instances, we have to pay attention to introduce "must-link" constraints since word instances might be distant from each other in the feature space even if they have the same sense. In addition, semi-supervised clustering method used in [23] is based on label propagation algorithm. Unlike this method, our proposed semi-supervised clustering approach is constraint-based with controlling the fluctuation of the centroid of a cluster. We could verify that this approach is effective in personal name disambiguation in Web search results [20]. Therefore, we refine this semi-supervised clustering approach suitable for word instances. Moreover, the recent supervised WSD systems described in Section 2.2 do not use information obtained from word instances clustered to seed instances although they add a lot of features to improve WSD accuracy. We believe that the accuracy of WSD can be improved by directly computing features from word instances clustered to seed instances. In this section, we give an overview of our system, describe our semi-supervised clustering approach for word instances, and explain how to compute features obtained from the clustering results.

3.1 System Architecture

Figure 1 illustrates our WSD system. This system extracts features for clustering and WSD, and performs semi-supervised clustering by introducing seed instances, as described in Section 3.2. After that, it computes features for WSD from the word instances in the generated clusters (Section 3.3). Using these features, a classifier can be constructed on the basis of three machine learning approaches: support vector machine (SVM), naïve Bayes (NB), and maximum entropy (ME).

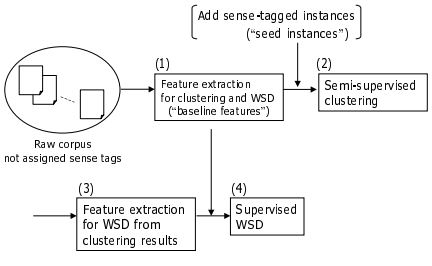


Fig. 1. Proposed WSD system

3.2 Semi-supervised Clustering

3.2.1 Features for Clustering

We use the following features:

- Morphological features
  - Bag-of-words (BOW), Part-of-speech (POS), and detailed POS classification. We extract these features from the target word itself and the two words to its right and left.

- Syntactic features
  - If the POS of a target word is a noun, extract the verb in a grammatical dependency relation with the noun.
  - If the POS of a target word is a verb, extract the noun in a grammatical dependency relation with the verb.
- Figures in Bunrui-Goi-Hyou (BGH)[1] [21]
  - 4 and 5 digits regarding the content word to the right and left of the target word. For example, when the target word is "syakai" ("society") and its left content word is "chiiki" ("community"), the figures of "chiiki" in thesaurus is "1.1720,4,1,3." We use 1172 and 11720 as 4 and 5 digits, respectively.
- 5 topics inferred on the basis of LDA [6]
  - We compute the log-likelihood of an instance using the "soft-tag"approach [3] where the topics are estimated from training data set (Fig.4) by regarding this set as unlabeled set using LDA.

We chose ChaSen[2] as the morphological analyzer and CaboCha[3] as the syntactic parser. We denote the feature vector $\boldsymbol{f}^x$ of word instance x as follows:

$$\boldsymbol{f}^x = (f_1^x, f_2^x, \cdots, f_n^x).$$

We refer to these features as "baseline features." We also use them in our WSD system (Section 3.3).

3.2.2 Semi-supervised Clustering

If the similarity between cluster C_{s_j} that contains seed instances and cluster C_i that does not contain seed instances is large, these two clusters are to be merged. However, when the distance between the centroids of these two clusters is large, the fluctuation of the centroid tends to be large. Therefore, when we merge a certain cluster C_i (its centroid vector $\boldsymbol{G}^{C_i}$) into C_{s_j} (its centroid vector $\boldsymbol{G}^{C_{s_j}}$) that contains seed instances, we first weight the feature vector $\boldsymbol{f}^x \in C_i$ relative to the distance between the centroids of the clusters. After that, we control the fluctuation of the centroid of a cluster by recomputing it with the weighted feature vectors. The details of the procedure are as follows:

We assume a cluster $C_{s_j}^{(k_j)}$ (number of elements: n_{s_j}) in which k_j clusters are merged and that contains a seed instance. We also assume that C_i (number of elements: n_i) is a cluster merged $(k_j + 1)$ times into $C_{s_j}^{(k_j)}$. We define the elements of cluster $C_{s_j}^{(0)}$ as the initial seed instances.

(1) Regarding each element contained in C_i that is merged into $C_{s_j}^{(k_j)}$, by using the distance $D(\boldsymbol{G}^{C_i}, \boldsymbol{G}^{C_{s_j}^{(k_j)}})$ between the centroid $\boldsymbol{G}^{C_{s_j}^{(k_j)}}$ of cluster $C_{s_j}^{(k_j)}$ and the centroid $\boldsymbol{G}^{C_i}$ of cluster C_i, we weight the feature vector $\boldsymbol{f}_{C_i}^{x_l}$ $(l = 1, \cdots, n_i)$ of word instances belonging to cluster C_i and define the generated cluster as $C_{i'}$ (number of elements: $n_{i'}$). The feature vector $\boldsymbol{f}_{C_{i'}}^{x_l}$ after weighting that belongs to cluster $C_{i'}$ is

[1] BGH is "Word List by Semantic Principles." In BGH, each word has a number called *a category number.*

[2] http://sourceforge.net/projects/masayu-a/

[3] http://sourceforge.net/projects/cabocha/

$$\boldsymbol{f}_{C_{i'}}^{x_l} = \frac{\boldsymbol{f}_{C_i}^{x_l}}{D(\boldsymbol{G}^{C_i}, \boldsymbol{G}^{C_{s_j}^{(k_j)}}) + c}, \tag{1}$$

where c is a constant to prevent the elements of $\boldsymbol{f}_{C_i}^{x_l}$ from being extremely large when $D(\boldsymbol{G}^{C_i}, \boldsymbol{G}^{C_{s_j}^{(k_j)}})$ is very close to 0. This value of c is set to 0.92 based on our preliminary experiments. We introduce adaptive Mahalanobis distance $D(\boldsymbol{G}^{C_i}, \boldsymbol{G}^{C_{s_j}^{(k_j)}})$, to overcome the drawback of the ordinary Mahalanobis distance whereby the covariance tends to be large when the number of elements in a cluster is small.

(2) We add the elements of $C_{i'}$ (number of elements: $n_{i'}$) to cluster $C_{s_j}^{(k_j)}$ (number of elements: n_{s_j}) that contain a seed instance and generate cluster $C_{s_j}^{(k_j+1)}$ (number of elements: $n_{s_j} + n_{i'}$) as follows:

$$C_{s_j}^{(k_j+1)} = \{\boldsymbol{f}_{C_{s_j}^{(k_j)}}^{x_1}, \cdots, \boldsymbol{f}_{C_{s_j}^{(k_j)}}^{x_{n_{s_j}}}, \boldsymbol{f}_{C_{i'}}^{x_1}, \cdots, \boldsymbol{f}_{C_{i'}}^{x_{n_{i'}}}\},$$

(3) The centroid $\boldsymbol{G}^{C_{s_j}^{(k_j+1)}}$ of cluster $C_{s_j}^{(k_j+1)}$ that merged with the $(k_j+1)^{th}$ cluster is defined as

$$\boldsymbol{G}^{C_{s_j}^{(k_j+1)}} = \frac{\sum_{\boldsymbol{f}^x \in C_{s_j}^{(k_j+1)}} \boldsymbol{f}^x}{n_{s_j} + n_{i'} \times \frac{1}{D(\boldsymbol{G}^{C_i}, \boldsymbol{G}^{C_{s_j}^{(k_j)}}) + c}}. \tag{2}$$

We weight the feature vector of the cluster to be merged in Equation (1), thus we also weight $n_{i'}$ as we can compute weighted average in Equation (2). If the cluster does not contain seed instances, the new centroid $\boldsymbol{G}^{new}$ of the cluster is computed using the following equation:

$$\boldsymbol{G}^{new} = \frac{\sum_{\boldsymbol{f}^x \in C_i} \boldsymbol{f}^x + \sum_{\boldsymbol{f}^x \in C_j} \boldsymbol{f}^x}{n_i + n_j}. \tag{3}$$

Figure 2 shows the semi-supervised clustering approach. Constraints between seed instances are also introduced at the beginning of the clustering in order to get accurate clustering results.

3.2.3 Seed Instances and Constraints for Clustering

To obtain higher clustering accuracy in semi-supervised clustering, it is important to introduce the initial seed instances and constraints between the seed instances properly. In this section, we describe how to introduce them in the semi-supervised clustering for word instances. We refer to a set of word instances for selecting seed instances as a "training data set." Generally, when we deal with word instances, it is important to consider the frequency of word senses in the training data set because there are some words whose instances are occupied by the small number of word senses, or other words whose instances are occupied by the large number of word senses. Thus, we consider this characteristic when we introduce seed instances for semi-supervised clustering. The number of training instances in our experiment was set to 100. The constraints between seed instances are "cannot-link" only, "must-link" only and both constraints. However, regarding "must-link" constraints, we have to exclude outlier instances. That is, if we select seed instances that contain outliers, the centroid of the initial cluster is not so accurate; inappropriate clusters tend to be generated in the subsequent clustering. If we

Algorithm: Semi-supervised clustering
Input: Set of feature vectors of word instances $\boldsymbol{f}^{x_i}$ $(i = 1, 2, \cdots, n)$ and seed instances $\boldsymbol{f}^{x_{s_j}}$ $(j = 1, 2, \cdots, u)$,
$E = \{\boldsymbol{f}^{x_1}, \boldsymbol{f}^{x_2}, \cdots, \boldsymbol{f}^{x_n}, \boldsymbol{f}^{x_{s_1}}, \boldsymbol{f}^{x_{s_2}}, \cdots, \boldsymbol{f}^{x_{s_u}}\}$.
Output: Set of clusters $\mathcal{C} = \{C_1, C_2, \cdots\}$ that contain the word instances that have the same sense.
Method:
1. Set feature vectors of each word instance $\boldsymbol{f}^{x_i}$ and each feature of seed instances $\boldsymbol{f}^{x_{s_j}}$ in E
as the initial cluster C_i and $C_{s_j}^{(k_j)}$, respectively.
$C_i = \{\boldsymbol{f}^{x_i}\}, C_{s_j}^{(k_j)} = \{\boldsymbol{f}^{x_{s_j}}\}$,
thus, the set of clusters $\mathcal{C} = \{C_1, C_2, \cdots, C_n, C_{s_1}^{(k_1)}, \cdots, C_{s_u}^{(k_u)}\}$,
where constraints are introduced between $C_{s_m}^{(k_m)}$ and $C_{s_n}^{(k_n)} (m \neq n)$.
k_h $(h = 1, \cdots, u) \leftarrow 0$,
where k_h denotes the frequency of merging other clusters into $C_{s_h}^{(k_h)}$.
2. **do**
2.1 Compute the similarity between C_i and C_j $(i \neq j)$, C_i and between $C_{s_h}^{(k_h)}$.
if the maximum similarity is obtained between C_i and $C_{s_h}^{(k_h)}$,
then compute the distance $D(\boldsymbol{G}^{C_i}, \boldsymbol{G}^{C_{s_h}^{(k_h)}})$
between the centroids $\boldsymbol{G}^{C_i}$ and $\boldsymbol{G}^{C_{s_h}^{(k_h)}}$ of C_i and $C_{s_h}^{(k_h)}$, respectively.
for $l = 1$ **to** n_{C_i} **do**
transform the feature vector $\boldsymbol{f}_{C_i}^{x_l}$ in C_i into $\boldsymbol{f}_{C_{i'}}^{x_l}$, by using Equation (1),
add $\boldsymbol{f}_{C_{i'}}^{x_l}$ to $C_{s_h}^{(k_h)}$
end
$k_h \leftarrow k_h + 1$
recompute the centroid $\boldsymbol{G}^{C_{s_h}^{(k_h)}}$ using Equation (2), and remove C_i from $\mathcal{C}$.
else if the maximum similarity is obtained between C_i and C_j,
then merge C_i and C_j to form a new cluster C^{new}, add C^{new} to $\mathcal{C}$, remove C_i and C_j from $\mathcal{C}$,
and recompute the centroid $\boldsymbol{G}^{new}$ of the cluster C^{new} by using Equation (3).
2.2 Compute similarities between C^{new} and all $C_i \in \mathcal{C}$ $(C_i \neq C^{new})$.
3. **until** All of the similarities computed in 2.2 between C_i and C_j are less than the predefined threshold.
4. **return** Set of clusters $\mathcal{C}$.

Fig. 2. Proposed semi-supervised clustering algorithm

exclude outliers, the centroid of the initial cluster becomes more accurate. We believe this idea leads to better clustering results. Figure 3 shows the algorithm and how to exclude outlier instances. We compute the new centroid generated by two clusters, then compute the distance between the new centroid and the clusters. If the distance is less than a predefined threshold, a "must-link" constraint is put between the two clusters. We compare the following two methods for selecting seed instances in semi-supervised clustering:

[Method I] Select seed instances for semi-supervised clustering from the whole training data set.

[Method II] First classify the training data set into each word sense. Then, considering the frequency of word sense, select seed instances for semi-supervised clustering from each classified word sense.

Figure 4 shows how to select seed instances from the data set of word instances.

Method I

We compared the following three ways of selecting seed instances for semi-supervised clustering:

Algorithm: Adding the "must-link" constraint that excludes outliers
Input: Set each feature vector of seed instance $\boldsymbol{f}^{x_{s_j}}$ $(j = 1, 2, \cdots, u)$,
$S = \{\boldsymbol{f}^{x_{s_1}}, \boldsymbol{f}^{x_{s_2}}, \cdots, \boldsymbol{f}^{x_{s_u}}\}$.
Output: Set of seed instances connected by "must-link" constraints
Method:
1. Set each feature of seed instance $\boldsymbol{f}^{x_{s_j}}$ as an initial cluster C_{s_j},
$C_{s_j} = \{\boldsymbol{f}^{x_{s_j}}\}$,
thus, the set of clusters $\mathcal{C} = \{C_{s_1}, \cdots, C_{s_u}\}$.
2. **do**
2.1 Find the cluster C_{s_w} that has the same sense as C_{s_v} $(v \neq w)$.
if C_{s_w} and C_{s_v} have different senses,
introduce a "cannot-link" constraint between them.
2.2 Compute the new centroid $\boldsymbol{G}^{new}$ based on clusters C_{s_v} and C_{s_w}.
2.3 Compute the distance $D(\boldsymbol{G}^{new}, \boldsymbol{G}^{C_{s_v}})$ between $\boldsymbol{G}^{new}$ and $\boldsymbol{G}^{C_{s_v}}$,
and the distance $D(\boldsymbol{G}^{new}, \boldsymbol{G}^{C_{s_w}})$ between $\boldsymbol{G}^{new}$ and $\boldsymbol{G}^{C_{s_w}}$.
2.4 **if** $D(\boldsymbol{G}^{new}, \boldsymbol{G}^{C_{s_v}}) < Th_{dis}$, and $D(\boldsymbol{G}^{new}, \boldsymbol{G}^{C_{s_w}}) < Th_{dis}$,
a "must-link" constraint is introduced between C_{s_v} and C_{s_w},
then merge them to form a new cluster C^{new}, add C^{new} to $\mathcal{C}$,
and remove C_{s_v} and C_{s_w} from $\mathcal{C}$.
else $D(\boldsymbol{G}^{new}, \boldsymbol{G}^{C_{s_v}}) > Th_{dis}$, $D(\boldsymbol{G}^{new}, \boldsymbol{G}^{C_{s_w}}) > Th_{dis}$
remove C_{s_w} from $\mathcal{C}$.
3. **until** $v = u$
4. **return** Initial seed clusters $\mathcal{C}$ with constraints.
(Outliers are excluded in clusters connected by "must-link" constraints.)

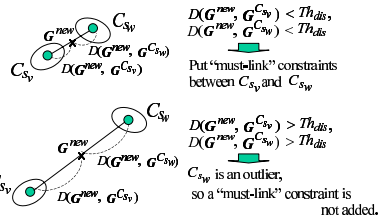


Fig. 3. Algorithm of excluding outlier word instances to add "must-link" constraint (left) and its overview (right)

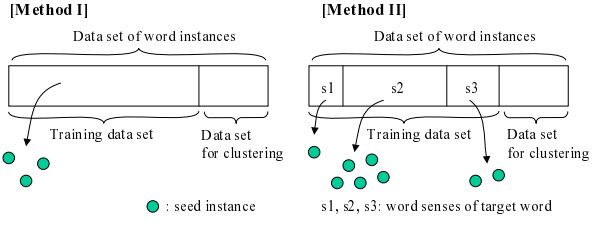


Fig. 4. How to select seed instances from the training data set

(I-1) Select initial seed instances randomly.
(I-2) Select initial seed instances on the basis of "KKZ" [10].
(I-3) As seed instances, select the centroid of a cluster generated by the K-means algorithm [14] whose initial instances are randomly selected (I-3rnd) or selected on the basis of KKZ (I-3KKZ).

"KKZ" in (I-2) is a cluster initialization method that select instances distant from each other [10]. In (I-1) and (I-2), we conduct experiments by introducing constraints of "cannot-link" only, "must-link" only, both constraints, and "cannot-link" and "must-link" without outliers. In (I-3), we introduce "cannot-link" constraints by simply assuming that the selected instances have different senses.

Method II

We compared the following cases: **(II-1)** select the seed instances by considering the frequency of word senses; and **(II-2)** select the seed instances in proportion to the frequency of word senses.

(II-1) We compared the following ways of selecting seed instances for semi-supervised clustering:

(II-1-1) Randomly select initial seed instances in order of word sense frequency,
(II-1-2) Select initial seed instances based on "KKZ" [10] in order of word sense frequency,
(II-1-3) First perform K-means clustering. Then set the centroids of the generated clusters as seed instances in order of word sense frequency for semi-supervised clustering. In this case, the initial instances for K-means clustering are either randomly selected (II-1-3rnd) or selected on the basis of KKZ (II-1-3KKZ).

As in Method I, in our experiment, we add constraints of "cannot-link" only, "must-link" only, both constraints, and "cannot-link" and "must-link" without outliers in (II-1-1) and (II-1-2), but only "cannot-link" constraints in (II-1-3).

(II-2) We use the D'Hondt method [17], a method for allocating seats to candidates in a proportional representation party list. The example in Figure 5 (left) assumes that parties A, B, and C gain votes of 1600, 700, and 300, respectively. When we allocate 10 seats to these parties, the seats are allocated in order of the value in parentheses. These figures are obtained by dividing votes by seat number. Parties A, B, and C gain 5, 4, and 1 seat, respectively. "Seat" and "party" correspond to "number of seed instances" and "word sense," respectively. Similarly, let us assume that word senses s1, s2, and s3, have 20, 50, and 15 instances, respectively (Fig. 5 (right)). When we select 10 instances, the seed instances are selected in order of the value in parentheses. We select 3, 5, and 2 seed instances from s1, s2, and s3, respectively.

As in (II-1), we compared the following three ways of selecting seed instances for semi-supervised clustering.

(II-2-1) Randomly select seed instances for semi-supervised clustering from each of the word senses selected using the D'Hondt method,
(II-2-2) Select seed instances for semi-supervised clustering on the basis of "KKZ" [10] from each of the word senses selected using the D'Hondt method,
(II-2-3) First select the initial instances randomly or by using KKZ for K-means clustering from each of the word senses selected using the D'Hondt method. Then set the centroids of the generated clusters as the seed instances for semi-supervised clustering. The initial instances for K-means clustering are either randomly selected (II-2-3rnd) or on the basis of KKZ (II-2-3KKZ).

In our experiment, we add constraints of "cannot-link" only, "must-link" only, both constraints, and "cannot-link" and "must-link" without outliers, but only "cannot-link" constraints in (II-2-3).

	Party A (1600)	Party B (700)	Party C (300)
seat 1 (/1)	1600 (1)	700 (3)	300 (8)
seat 2 (/2)	800 (2)	350 (6)	150
seat 3 (/3)	533 (4)	233 (9)	
seat 4 (/4)	400 (5)	175 (10)	
seat 5 (/5)	320 (7)		

D'Hondt method

	s2 (50)	s1 (20)	s3 (15)
seed 1 (/1)	50 (1)	20 (3)	15 (5)
seed 2 (/2)	25 (2)	10 (8)	8 (9)
seed 3 (/3)	17 (4)	7 (10)	3
seed 4 (/4)	13 (6)		
seed 5 (/5)	10 (7)		

* s1, s2, s3: word senses
Selecting word senses using D'Hondt method

Fig. 5. D'Hondt method and its application to our system

3.3 Word Sense Disambiguation

3.3.1 Features Obtained Using Clustering Results

We add features obtained from the clustering results to the "baseline features" described in Section 3.2.1 for WSD. Word instances in the generated clusters are aggregated on the basis of their similarity to the seed instances. Therefore, we expect that we can obtain features such as context information from the generated clusters. In particular, we compute features for WSD from the generated clusters. We believe that these features will contribute to the accuracy of WSD. We extracted features from:

(a) inter-cluster information,
(b) context information regarding adjacent words $w_i w_{i+1}$, $(i = -2, \cdots, 1)$, and
(c) context information regarding two words to the right and left of the target word, $w_{-2} w_{-1} w_0 w_{+1} w_{+2}$.

Features (b) and (c) are often used to extract collocations. We use them as features that reflect the concept of "one sense per collocation" [22].

Regarding (a), we employ the term frequency (TF) in a cluster, cluster ID (CID), and the sense frequency (SF) of seed instances. If the values of TF to the right and left of the target word are large, its word sense can be easily identified. Moreover, each generated cluster aggregates similar word instances. Thus, if we use the CID as features for WSD, we can obtain an effect equivalent to assigning the correct word sense. Furthermore, our semi-supervised clustering uses seed instances with sense tags. Therefore, if we use SF as a feature, the word sense of the target word can be easily determined. TF and SF are normalized by the total number of terms and seed instances in each cluster, respectively.

Table 1. two-by-two contingency table showing the dependence of occurrences of w_i and w_{i+1}.

	w_i	$\neg w_{i+1}$
w_{i+1}	O_{11}	O_{12}
$\neg w_{i+1}$	O_{21}	O_{22}

Regarding (b), we compute mutual information (MI), T-score (T), and χ^2 ($CHI2$) for adjacent words. MI is defined as

$$MI = \log \frac{p(w_i, w_{i+1})}{p(w_i)p(w_{i+1})},$$

where $p(w_i)$ and $p(w_i, w_{i+1})$ are the probability of occurence of w_i and the probability of the co-occurrence of w_i and w_{i+1}. T-score is defined as

$$T = \frac{p(w_i, w_{i+1}) - p(w_i)p(w_{i+1})}{\sqrt{s^2/N}},$$

where s^2 and N are sample variance and sample size, respectively. Based on the 2-by-2 contingency table (Table 1), $CHI2$ is defined as

$$CHI2 = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})}.$$

In (c), we employ information gain regarding two words to the right and left of the target word. We first compute the entropy by using the set D of feature vectors of word instance as

$$entropy(D) = -\sum_{j=1}^{|s_j|} P_r(s_j) log_2 P_r(s_j), \tag{4}$$

where s_j, $|s_j|$ and $P_r(s_j)$ are word sense, the number of word senses, and its probability of occurrence, respectively. Then, using Equation (5), we compute the entropy of w_i after the clusters are generated:

$$entropy_{w_i}(D) = \sum_{j=1}^{|\nu|} \frac{|s_j|}{|D|} entropy(D), \tag{5}$$

where $|\nu|$ is the number of generated clusters. Using Equations (4) and (5), the information gain $IG(w_0)$ for target word w_0 is defined as

$$IG(w_0) = entropy(D) - entropy_{w_0}(D).$$

Finally, by considering the context for two words to the right and left of the target word w_0, the information gain for w_0 is computed as follows:

$$IG(w_0) = \sum_{i=-2}^{2} IG(w_i).$$

These features for seed instances are also computed in order to verify WSD accuracy.

4 Experiments

4.1 Experimental Data

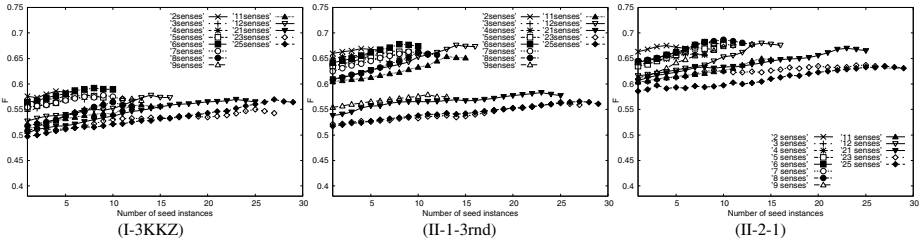
We used the RWC corpus from the "SENSEVAL-2 Japanese Dictionary Task" [11]. In this corpus, sense tags were manually assigned to 3,000 Japanese newspaper (Mainichi Shimbun) articles issued in 1994. The sense tags were assigned to 148,558 ambiguous words that had headwords in a Japanese dictionary (Iwanami Kokugo Jiten) [15] and whose POS was either noun, verb, or adjective. We used the same 100 target words (50 nouns and 50 verbs) as in the SENSEVAL-2 Japanese Dictionary Task.

4.2 Semi-supervised Clustering

In this experiment, we first introduce seed instances and constraints as described in Section 3.2.3. Seed instances are selected from the training data set that corresponds to 80% of the data set of word instances, and test data set for clustering corresponds to 20% of the data set of word instances. The clustering results shown in Section 4.2.2 are based on 5-fold cross validation.

4.2.1 Evaluation Measure

We evaluated the accuracy of our semi-supervised clustering based on F, i.e., the harmonic mean of "*purity*" and "*inverse purity*" [2].



(I-3KKZ) (II-1-3rnd) (II-2-1)

Fig. 6. Clustering accuracy obtained from (I-3KKZ), (II-1-3rnd), and (II-2-1)

Table 2. Comparison of clustering accuracies (F)

	Method I	Method (II-1)	Method (II-2)
Proposed method	**0.543** (I-3KKZ)	**0.592** (II-1-3rnd)	**0.646** (II-2-1)
Bar-Hillel et al. [1]	0.516	0.570	0.608
Xing et al. [9]	0.494	0.539	0.591
Klein et al. [5]	0.448	0.504	0.570
Fixed centroid	0.385	0.402	0.514
Agglomerative clustering	0.380	0.389	0.471

4.2.2 Experimental Results

Because of space limitations, we only show the best clustering results for Methods I, II-1, and II-2. We attempted to add constraints of (a) "cannot-link" only, (b) "must-link" only, (c) both constraints, and (d) "cannot-link" and "must-link" without outliers. Figure 6 shows clustering results when we add constraint (d) because we found that the best clustering accuracy is obtained by using this constraint. These results are obtained using I-3KKZ in Method I, II-1-3rnd in Method II-1, and II-2-1 in Method II-2. In these graphs, each line shows the clustering accuracy obtained for words of each number of word senses. Some of the number of word senses are absent (e.g., 10, 13-20, 22, 24) because such ambiguous words do not exist. The number of seed instances was from one to four plus the original number of word senses defined in the dictionary [15]. Table 2 summarizes the clustering accuracy F obtained by our semi-supervised clustering, distance-based semi-supervised clustering reviewed in Section 2.1, clustering in the case that the centroid of a cluster is fixed, and ordinary agglomerative clustering. These F values are average results for the case of two seed instances in addition to the original word senses, since this number gave the best clustering accuracy.

4.2.3 Discussion

Regarding Method I, the best clustering accuracy is obtained for the centroid of a cluster generated by the K-means algorithm whose initial instances were selected on the basis of KKZ as the seed instances for semi-supervised clustering. When the seed instances are selected from the whole set of training data set, the representative instances tend to be selected by K-means clustering after selecting distant initial instances on the basis of KKZ. Regarding Method II, II-2, which selects seed instances in proportion to the frequency of word senses, is more effective than II-1 which selects seed instances by considering the frequency of word senses. In particular, we found that randomly selecting seed instances is more effective than selecting them by KKZ for seed instances

in the same word senses. In addition, we could obtain the best clustering accuracy in II-2-1 among all of our experiments. From the results, we found that it is effective to take into account the frequency distribution in selecting seed instances.

As described in Section 4.2.2, in most cases, we found that the best clustering accuracy is obtained when two more seed instances are added to the original number of word senses. Although these seed instances are sense-tagged ones, we consider that, in the clustering process, such extra seed instances contribute to discovering new word senses that are not defined in a dictionary by applying semi-supervised clustering to word instances.

According to the results in Table 2, our semi-supervised clustering outperforms other distance-based approaches. We believe that it is better because it locally adjusts the centroid of a cluster whereas the other distance-based semi-supervised clustering approaches transform the feature space globally.

4.3 Word Sense Disambiguation

In order to verify WSD accuracy, we also compute the features described in Section 3.3.1 for sense-tagged training data.

4.3.1 Evaluation Measure

We employ "accuracy" as an evaluation measure for WSD. This measure is based on "fine-grained scoring" that judges the right answer when the word sense that the system outputs completely corresponds to a predefined correct word sense.

4.3.2 Experimental Results

We constructed classifiers using the features described in Section 3.2.1 and 3.3.1 and conducted experiments using five-fold cross validation. Table 3 shows the experimental results for our WSD system (OURS) and for features employed by the participants (CRL, TITECH, NAIST) of the SENSEVAL-2 Japanese Dictionary task. "OURS" means using the baseline features described in Section 3.2.1.

4.3.3 Discussion

For each machine learning approach (SVM, NB, and ME), our WSD had the best accuracy when we added features from clustering results, especially CID, MI and IG, to the

Table 3. WSD accuracies

Features	SVM	NB	ME	Features	SVM	NB	ME
OURS (not clustered)	0.663	0.667	0.662	CRL (not clustered)	0.775	0.778	0.773
OURS + MI (not clustered)	0.666	0.669	0.664	CRL + MI (not clustered)	0.776	0.780	0.775
OURS + CID + MI + IG	**0.780**	**0.782**	**0.779**	CRL + CID + MI + IG	**0.778**	**0.783**	**0.780**
OURS + CID + T + IG	0.768	0.777	0.764	CRL + CID + T + IG	0.778	0.779	0.777
OURS + CID + CHI2 + IG	0.762	0.765	0.757	CRL + CID + CHI2 + IG	0.776	0.779	0.775
TITECH (not clustered)	0.661	0.663	0.660	NAIST (not clustered)	0.745	0.747	0.743
TITECH + MI (not clustered)	0.663	0.665	0.662	NAIST + MI (not clustered)	0.747	0.748	0.745
TITECH + CID + MI + IG	**0.767**	**0.770**	**0.764**	NAIST + CID + MI + IG	**0.765**	**0.767**	**0.764**
TITECH + CID + T + IG	0.765	0.767	0.759	NAIST + CID + T + IG	0.756	0.760	0.755
TITECH + CID + CHI2 + IG	0.756	0.759	0.751	NAIST + CID + CHI2 + IG	0.752	0.754	0.747

baseline features. Among the features (a) (see Section 3.3.1), we found that CID contributed to improvement in WSD accuracy compared with TF and SF. Moreover, among the features (b) (see Section 3.3.1), MI was more effective, and T and $CHI2$ were not so effective. This shows that word instances that have similar contexts can be aggregated into seed instances in the generated clusters. Although our method, TITECH, and NAIST use simple features such as the BOW of the target word, POS, and so on, WSD accuracy was significantly improved by adding features computed from clustering results. For these systems, we obtained 0.020 to 0.117 improvement compared with results for which clustering was not performed. This indicates that the information required for WSD is complemented by adding features computed from clustering results. On the other hand, for the CRL system, we obtained only a 0.003 to 0.007 improvement relative to the results for which clustering was not performed. The CRL system achieve high WSD accuracy using a lot of features. Therefore, we consider that, even if more features are added to original features, they are not so effective to improve WSD accuracy significantly.

5 Conclusion

We verified how a semi-supervised clustering approach contributes to word sense disambiguation (WSD). We found that method II-2-1 that selects the word sense by using the D'Hondt method and randomly selects seed instances from ones that belong to the word sense is effective in semi-supervised clustering for word instances. We also found that the accuracy of WSD is improved by constructing a classifier using features such as CID, MI, and IG obtained from semi-supervised clustering results. In the future, we plan to develop a much more accurate semi-supervised clustering approach and look for features that can lead to higher accuracy for WSD.

References

1. Bar-Hillel, A., Hertz, T., Shental, N.: Learning Distance Functions Using Equivalence Relations. In: Proc. of the 20th International Conference on Machine Learning (ICML 2003), pp. 577–584 (2003)
2. Hotho, A., Nürnberger, A., Paaß, G.: A Brief Survey of Text Mining. GLDV-Journal for Computational Linguistics and Language Technology 20(1), 19–62 (2005)
3. Cai, J.F., Lee, W.S., Teh, Y.W.: Improving Word Sense Disambiguation Using Topic Features. In: Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP 2007), pp. 1015–1023 (2007)
4. Davidov, D., Rappoport, A.: Classification of Semantic Relationships betwen Nominals Using Pattern Clusters. In: Proc. of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2008:HLT), pp. 227–235 (2008)
5. Klein, D., Kamvar, S.D., Manning, C.D.: From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering. In: Proc. of the 19th International Conference on Machine Learning (ICML 2002), pp. 307–314 (2002)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)

7. Agirre, E., Ansa, O., Hovy, E., Martínez, D.: Enriching Very Large Ontologies Using the WWW. In: Proc. of 1st International Workshop on Ontology Learning (OL 2000). Held in Conjunction with the 14th European Conference on Artificial Intelligence (ECAI 2000) (2000)
8. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: The 90% Solution. In: Proc. of the Human Language Technology Conference of the North American Chapter of the ACL (HLT-NAACL 2006), pp. 57–60 (2006)
9. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.J.: Distance Metric Learning with Application to Clustering with Side-Information. Advances in Neural Information Processing Systems 15, 521–528 (2003)
10. Katsavounidis, I., Kuo, C., Zhang, Z.: A New Initialization Technique for Generalized Lloyd Iteration. IEEE Signal Processing Letters 1(10), 144–146 (1994)
11. Shirai, K.: SENSEVAL-2 Japanese Dictionary Task. In: Proc. of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2), pp. 33–36 (2001)
12. Wagstaff, K., Cardie, C.: Clustering with Instance-level Constraints. In: Proc. of the 17th International Conference on Machine Learning (ICML 2000), pp. 1103–1110 (2000)
13. Wagstaff, K., Rogers, S., Schroedl, S.: Constrained K-means Clustering with Background Knowledge. In: Proc. of the 18th International Conference on Machine Learning (ICML 2001), pp. 577–584 (2001)
14. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proc. of the 5th Berkeley Symposium on Mathmatical Statistics and Probability, pp. 281–297 (1967)
15. Nishio, M., Iwabuchi, E., Mizutani, S.: Iwanami Kokugo Jiten Dai Go Han. Iwanami Shoten (1994) (in Japanese)
16. Muggleton, S.: Inductive Logic Programming. New Generation Computing 8(4), 295–318 (1991)
17. Taagepera, R., Shugart, M.S.: Seats and Votes: The Effects and Determinants of Electoral Systems. Yale University Press (1991)
18. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised Clustering by Seeding. In: Proc. of the 19th International Conference on Machine Learning (ICML 2002), pp. 27–34 (2002)
19. Specia, L., Stevenson, M., Nunes, M.G.V.: Learning Expressive Models for Word Sense Disambiguation. In: Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007), pp. 41–48 (2007)
20. Sugiyama, K., Okumura, M.: Personal Name Disambiguation in Web Search Results Based on a Semi-supervised Clustering Approach. In: Goh, D.H.-L., Cao, T.H., Sølvberg, I.T., Rasmussen, E. (eds.) ICADL 2007. LNCS, vol. 4822, pp. 250–256. Springer, Heidelberg (2007)
21. The National Language Research Institute. Bunrui Goi Hyou. Shuueisha (1994) (in Japanese)
22. Yarowsky, D.: Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In: Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL 1995), pp. 189–196 (1995)
23. Niu, Z.-Y., Ji, D.-H., Tan, C.L.: A Semi-Supervised Feature Clustering Algorithm with Application to Word Sense Disambiguation. In: Proc. of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), pp. 907–914 (2005)
24. Zhong, Z., Ng, H.T., Chan, Y.S.: Word Sense Disambigation Using OntoNotes: An Empirical Study. In: Proc. of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), pp. 1002–1010 (2008)