# JAIST: Clustering and Classification based Approaches for Japanese WSD

**Kiyoaki Shirai** **Makoto Nakamura**
Japan Advanced Institute of Science and Technology
{kshirai,mnakamur}@jaist.ac.jp

## Abstract

This paper reports about our three participating systems in SemEval-2 Japanese WSD task. The first one is a clustering based method, which chooses a sense for, not individual instances, but automatically constructed clusters of instances. The second one is a classification method, which is an ordinary SVM classifier with simple domain adaptation techniques. The last is an ensemble of these two systems. Results of the formal run shows the second system is the best. Its precision is 0.7476.

## 1 Introduction

This paper reports about our systems in SemEval-2 Japanese Word Sense Disambiguation (WSD) task (Okumura et al., 2010). This task is a lexical sample task for Japanese WSD and has the following two characteristics. First, a balanced word-sense tagged corpus is used for the task. Since it consists of sub-corpora of several domains or genres, domain adaptation might be required. Second, the task takes into account not only the instances having a sense in the given set but also the instances having a sense not found in the set (called 'new sense'). Participants are required to identify new senses of words in this task.

The second characteristics of the task is mainly considered in our system. A clustering based approach is investigated to identify new senses. Our system first constructs a set of clusters of given word instances using unsupervised clustering techniques. This is motivated by the fact that the new sense is not defined in the dictionary, and sense induction without referring to the dictionary would be required. Clusters obtained would be sets of instances having the same sense, and some of them would be new sense instances. Then each cluster is judged whether instances in it have a new sense or not. An ordinary classification-based approach is also considered. That is, WSD classifiers are trained by a supervised learning algorithm. Furthermore, simple techniques considering genres of sub-corpora are incorporated into both our clustering and classification based systems.

The paper continues as follows, Section 2 describes our three participating systems, JAIST-1, JAIST-2 and JAIST-3. The results of these systems are reported and discussed in Section 3. Finally we conclude the paper in Section 4.

## 2 Systems

### 2.1 JAIST-1: Clustering based WSD System

JAIST-1 was developed by a clustering based method. The overview of the system is shown in Figure 1. It consists of two procedures: (A) clusters of word instances are constructed so that the instances of the same sense are merged, (B) then similarity between a cluster and a sense in a dictionary is measured in order to determine senses of instances in each cluster.

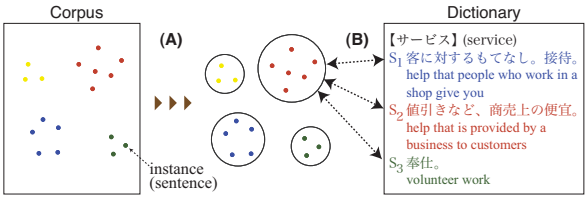


Figure 1: Overview of JAIST-1

#### 2.1.1 Clustering of Word Instances

As previous work applying clustering techniques for sense induction (Schütze, 1998; Agirre and Soroa, 2007), each instance is represented by a feature vector. In JAIST-1, the following 4 vectors are used for clustering.

**Collocation Vector** This vector reflects collocation including the target instance. Words or POSs appearing just before and after the target instance are used as features, i.e. they correspond to one dimension in the vector. The weight of each feature is 1 if the feature exists for the instance, or 0 if not.

**Context Vector** The vector reflects words in the context of the target instance. All content words appearing in the context are used as features. The window size of the context is set to 50. Furthermore, related words are also used as features to en-



*Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, pages 379–382,
Uppsala, Sweden, 15-16 July 2010. 

rich the information in the vector. Related words are defined as follows: first topics of texts are automatically derived by Latent Dirichlet Allocation (LDA) (Blei et al., 2003), then words which are the most closely associated with each topic are formed into a 'related word set'. If one word in a related word set appears in the context, other words in that set also have a positive weight in the vector. More concretely, the weight of each feature is determined to be 1 if the word appears in the context or 0.5 if the word does not appear but is in the related word set.

**Association Vector** Similarly to context vector, this reflects words in the context of the target instance, but data sparseness is alleviated in a different manner. In advance, the co-occurrence matrix $A$ is constructed from a corpus. Each row and column in $A$ corresponds to one of the most frequent 10,000 content words. Each element $a_{i,j}$ in the matrix is $P(w_i|w_j)$, conditional probability representing how likely it is that two words $w_i$ and $w_j$ will occur in the same document. Now $j$-th column in $A$ can be regarded as the co-occurrence vector of $w_j$, $\vec{o}(w_j)$. Association vector is a normalized vector of sum of $\vec{o}(w_j)$ for all words in the context.

**Topic Vector** Unlike other vectors, this vector reflects topics of texts. The topics $z_j$ automatically derived by PLSI (Probabilistic Latent Semantic Indexing) are used as features. The weight for $z_j$ in the vector is $P(z_j|d_i)$ estimated by Folding-in algorithm (Hofmann, 1999), where $d_i$ is the document containing the instance. Topic vector is motivated by the well-known fact that word senses are highly associated with the topics of documents.

Target instances are clustered by the agglomerative clustering algorithm. Similarities between instances are calculated by cosine measure of vectors. Furthermore, pairs of instances in different genre sub-corpora are treated as 'cannot-link', so that they will not be merged into the same cluster. Clustering procedure is stopped when the number of instances in a cluster become more than a threshold $N_c$. $N_c$ is set to 5 in the participating system.

The clustering is performed 4 times using 4 different feature vectors. Then the best one is chosen from the 4 sets of clusters obtained. A set of cluster $C$ (=$\{C_i\}$) is evaluated by $E(C)$

$$E(C) = \textstyle\sum_i coh(C_i) \quad (1)$$

where 'cohesiveness' $coh(C_i)$ for each cluster $C_i$ is defined by (2).

$$\begin{aligned} coh(C_i) &= \frac{1}{|C_i|}\sum_{j=1}^{|C_i|} rel\text{-}sim(\vec{v}_{ij}, \vec{g}_i) \\ &= \frac{1}{|C_i|}\sum_{j=1}^{|C_i|} \frac{sim(\vec{v}_{ij}, \vec{g}_i)}{\max_j sim(\vec{v}_{ij}, \vec{g}_i)} \end{aligned} \quad (2)$$

$\vec{v}_{ij}$ is an instance vector in the cluster $C_i$, while $\vec{g}_i$ is an average vector of $C_i$. $rel\text{-}sim(\vec{v}_{ij}, \vec{g}_i)$ means the relative similarity between the instance vector and average vector. Intuitively, $coh(C_i)$ evaluates how likely instances in the cluster are similar each other. $C$ such that $E(C)$ is maximum is chosen as the final set of clusters.

### 2.1.2 Similarity between Clusters and Senses

After clustering, similarity between a cluster $C_i$ and a sense $S_j$ in the dictionary, $sim(C_i, S_j)$, is calculated for WSD. $C_i$ and $S_j$ are represented by cluster vector $\vec{c}_i$ and sense vector $\vec{s}_j$, respectively. Then cosine measure between these two vectors is calculated as $sim(C_i, S_j)$.

The cluster vector $\vec{c}_i$ is defined as (3):

$$\vec{c}_i = \frac{1}{N}\sum_{e_{ik}\in C_i}\sum_{t_l \in e_{ik}} \vec{o}(t_l) \quad (3)$$

In (3), $e_{ik}$ stands for an instance in the cluster $C_i$, $t_l$ words appearing in the context of $e_{ik}$, $\vec{o}(t_l)$ co-occurrence vector of $t_l$ (similar one used in association vector), and $N$ the constant for normalization. So $\vec{c}_i$ is similar to association vector, but the co-occurrence vectors of words in the contexts of all instances in the cluster are summed.

The sense vector $\vec{s}_j$ is defined as in (4).

$$\vec{s}_j = \frac{1}{N}\left(\sum_{t_k \in D_j} \vec{o}(t_k) + \sum_{t_l \in E_j} w_e \cdot \vec{o}(t_l)\right) \quad (4)$$

$D_j$ stands for definition sentences of the sense $S_j$ in the Japanese dictionary Iwanami Kokugo Jiten (the sense inventory in this task), while $E_j$ a set of example sentences of $S_j$. Here $E_j$ includes both example sentences from the dictionary and ones excerpted from a sense-tagged corpus, the training data of this task. $w_e$ is the parameter putting more weight on words in example sentences than in definition sentences. We set $w_e = 2.0$ through the preliminary investigation.

Based on $sim(C_i, S_j)$, the system judges whether the cluster is a collection of new

sense instances. Suppose that $MaxSim_i$ is $\max_j sim(C_i, S_j)$, the maximum similarity between the cluster and the sense. If $MaxSim_i$ is small, the cluster $C_i$ is not similar to any defined senses, so instances in $C_i$ could have a new sense. The system regards that the sense of instances in $C_i$ is new when $MaxSim_i$ is less than a threshold $T_{ns}$. Otherwise, it regards the sense of instances in $C_i$ as the most similar sense, $S_j$ such that $j = \arg\max_j sim(C_i, S_j)$.

The threshold $T_{ns}$ for each target word is determined as follows. First the training data is equally subdivided into two halves, the development data $D_{dev}$ and the training data $D_{tr}$. Next, JAIST-1 is run for instances in $D_{dev}$, while example sentences in $D_{tr}$ are used as $E_j$ in (4) when sense vectors are constructed. For words where new sense instances exist in $D_{dev}$, $T_{ns}$ is optimized for the accuracy of new sense detection. For words where no new sense instances are found in $D_{dev}$, $T_{ns}$ is determined by the minimum of $MaxSim_i$ as follows:

$$T_{ns} = (\min_i MaxSim_i) \times \gamma \quad (5)$$

Since even the cluster of which $MaxSim_i$ is minimum represents not a new but a defined sense, the minimum of $MaxSim_i$ is decreased by $\gamma$. To determine $\gamma$, the ratios

$$\frac{MaxSim_i \text{ of clusters of new senses}}{MaxSim_i \text{ of clusters of defined senses}} \quad (6)$$

are investigated for 5 words[1]. Since we found the ratios are more than 0.95, we set $\gamma$ to 0.95.

### 2.2 JAIST-2: SVM Classifier with Simple Domain Adaptation

Our second system JAIST-2 is the classification based method. It is a WSD classifier trained by Support Vector Machine (SVM). SVM is widely used for various NLP tasks including Japanese WSD (Shirai and Tamagaki, 2004). In this system, new sense is treated as one of the sense classes. Thus it would never choose "new sense" for any instances when no new sense instance is found in the training data. We used the LIBSVM package[2] to train the SVM classifiers. Linear kernel is used with default parameters.

The following conventional features of WSD are used for training the SVM classifiers.

- $W(0), W(-1), W(-2), W(+1), W(+2)$
  $P(-1), P(-2), P(+1), P(+2)$

  Words and their POSs appearing before or after a target instance. A number in parentheses indicates the position of a word from a target instance. $W(0)$ means a target instance itself.

- $W(-2)\&W(-1)$,$W(+1)\&W(+2)$,$W(-1)\&W(+1)$
  $P(-2)\&P(-1), P(+1)\&P(+2), P(-1)\&P(+1)$

  Pairs of words (or their POSs) near a target instance.

- Base form of content words appearing in the context (bag-of-words).

The data used in this task is a set of documents with 4 different genre codes: OC (Web page), OW (white paper), PB (book) and PN (newspaper). The training data consists of documents of 3 genres OW, PB and PN, while the test data contains all 4 genres. Considering domain adaptation, each feature $f_i$ is represented as $f_i + g$ when SVM classifiers are trained. $g$ is one of the genre codes $\{OW, PB, PN\}$ if $f_i$ is derived from the documents of only one genre $g$ in the training data, otherwise $g$ is 'multi'. For instances in the test data, only features $f_i+g_t$ and $f_i+multi$ are used, where $g_t$ is the genre code of the document of the target instance. If $g_t$ is OC (which is not included in the training data), however, all features are used. The above method aims at distinguishing genre intrinsic features and improving the WSD performance by excluding features which might be associated with different genres.

### 2.3 JAIST-3: Ensemble of Two Systems

The third system combines clustering based method (JAIST-1) and classification based method (JAIST-2). The basic idea is that JAIST-1 be used only for reliable clusters, otherwise JAIST-2 is used. Here 'reliable cluster' means a cluster such that $MaxSim_i$ is high. The greater the similarity between the cluster and the sense is, the more likely the chosen sense is correct. Furthermore, JAIST-1 is used for new sense detection. The detailed procedure in JAIST-3 is:

1. If JAIST-1 judges a cluster to be a collection of new sense instances, output 'new sense' for instances in that cluster.
2. For instances in the top $N_{cl}$ clusters of $MaxSim_i$,output senses chosen by JAIST-1.
3. Otherwise output senses chosen by JAIST-2.

[1] Among 50 target words in this task, there exist new sense instances of only 'kanou'(possibility) in $D_{dev}$. So we checked 4 more words, other than target words.

[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

For the optimization of $N_{cl}$, $D_{dev}$ and $D_{tr}$, each is a half of the training data described in Subsection 2.1, are used. $D_{tr}$ is used for training SVM classifiers (JAIST-2). Then $N_{cl}$ is determined so that the precision of WSD on $D_{dev}$ is optimized. In the participating system, $N_{cl}$ is set to 1.

## 3 Evaluation

Table 1 shows the results of our participating systems and the baseline system MFS, which always selects the most frequent sense in the training data. The column WSD reveals the precision (P) of word sense disambiguation, while the column NSD shows accuracy (A), precision (P) and recall (R) of new sense detection.

Table 1: Results

| | WSD | NSD | | |
|---|---|---|---|---|
| | P | A | P | R |
| MFS | 0.6896 | 0.9844 | 0 | 0 |
| JAIST-1 | 0.6864 | 0.9512 | 0.0337 | 0.0769 |
| JAIST-2 | 0.7476 | 0.9872 | 1 | 0.1795 |
| JAIST-3 | 0.7208 | 0.9532 | 0.0851 | 0.2051 |

JAIST-1 is the clustering based method. Performance of the clustering is also evaluated: Purity was 0.9636, Inverse-Purity 0.1336 and F-measure 0.2333. Although this system was designed for new sense detection, it seems not to work well. It could correctly find only three new sense instances. The main reason is that there were few instances of the new sense in the test data. Among 2,500 instances (50 instances of each word, for 50 target word), only 39 instances had the new sense. Our system supposes that considerable number of new sense instances exist in the corpus, and tries to gather them into clusters. However, JAIST-1 was able to construct only one cluster containing multiple new sense instances. The proposed method is inadequate for new sense detection when the number of new sense instances is quite small.

For domain adaptation, features which are intrinsic to different genres were excluded for test instances in JAIST-2. When we trained the system using all features, its precision was 0.7516, which is higher than that of JAIST-2. Thus our method does not work at all. This might be caused by removing features that were derived from different genre sub-corpora, but effective for WSD. More sophisticated ways to remove ineffective features would be required.

JAIST-3 is the ensemble of JAIST-1 and JAIST-2. Although a little improvement is found by combining two different systems in our preliminary experiments, however, the performance of JAIST-3 was worse than JAIST-2 because of the low performance of JAIST-1. We compared WSD precision of three systems for 50 individual target words, and found that JAIST-2 is almost always the best. The only exceptional case was the target word 'ookii'(big). For this adjective, the precision of JAIST-1, JAIST-2 and JAIST-3 were 0.74, 0.16 and 0.18, respectively. The precision of SVM classifiers (JAIST-2) is quite bad because of the difference of text genres. All 50 test instances of this word were excerpted from Web sub-corpus, which was not included in the training data. Furthermore, word sense distributions of test and training data were totally different. JAIST-1 works better in such a case. Thus clustering based method might be an alternative method for WSD when sense distribution in the test data is far from the training data.

## 4 Conclusion

The paper reports the participating systems in SemEval-2 Japanese WSD task. Clustering based method was designed for new sense detection, however, it was ineffective when there were few new sense instances. In future, we would like to examine the performance of our method when it is applied to a corpus including more new senses.

## References


Eneko Agirre and Aitor Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 7–12.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the SIGIR*, pages 50–57.

Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono. 2010. Semeval-2010 task: Japanese WSD. In *Proceedings of the SemEval-2010: 5th International Workshop on Semantic Evaluations*.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Kiyoaki Shirai and Takayuki Tamagaki. 2004. Word sense disambiguation using heterogeneous language resources. In *Proceedings of the First IJCNLP*, pages 614–619.