Division of Example Sentences Based on the Meaning of a Target Word Using Semi-supervised Clustering

Hiroyuki Shinnou, Minoru Sasaki

Ibaraki University,
4-12-1 Nakanarusawa, Hitachi, Ibaraki, Japan 316-8511
{shinnou, msasaki}@mx.ibaraki.ac.jp

Abstract

In this paper, we describe a system that divides example sentences (data set) into clusters, based on the meaning of the target word, using a semi-supervised clustering technique. In this task, the estimation of the cluster number (the number of the meaning) is critical. Our system primarily concentrates on this aspect. First, a user assigns the system an initial cluster number for the target word. The system then performs general clustering on the data set to obtain small clusters. Next, using constraints given by the user, the system integrates these clusters to obtain the final clustering result. Our system performs this entire procedure with high precision and requiring only a few constraints. In the experiment, we tested the system for 12 Japanese nouns used in the SENSEVAL2 Japanese dictionary task. The experiment proved the effectiveness of our system. In the future, we will improve sentence similarity measurements.

1. Introduction

We perform the task of collecting example sentences from a corpus, based on the meaning of the target word. It is simple to extract example sentences that include the target word. The hurdle is to divide these sentences into clusters based on the meaning of the target word. This paper describes our system, which efficiently overcomes this hurdle.

Example sentences that are clustered based on the meaning of the target word, are useful for full-scale semantic analysis. For example, we can design a classifier to solve word sense disambiguation, using example sentences as training data during inductive learning (Masaki Murata and Masao Utiyama and Kiyotaka Uchimoto and Qing Ma and Hitoshi Isahara, 2001). We can easily construct the case slot of a verb (Philip, 1992) using example sentences clustered based on the meaning of the verb. A thesaurus can easily be designed (Hindle, 1990) using example sentences clustered based on the meaning of a noun.

Clustering of example sentences based on the meaning of a target word can be performed by distinguishing between the different senses of in an example sentence. In other words, this task is termed as word sense disambiguation. This task may be performed using a (semi-) supervised learning approach. However, this approach is not practical for two problems. The first problem is the cost of constructing the training data. Supervised learning requires a large amount of training data. If there are many target words, it is impossible to construct training data corresponding to each word. The second problem is the definition of the senses of a target word. When using a (semi-) supervised learning approach, it is necessary to define the senses of the target word in advance. It is difficult to maintain a uniform sense granularity, and a minor sense may easily be overlooked.

For this task, unsupervised learning, i.e., clustering, is available. However, several clustering methods, such as k-means, require the number of clusters, i.e., the number of meanings of the target word. These methods cannot be used for our task. Some clustering methods can estimate the number of clusters; however, this estimation is essentially impossible because it fixes the sense granularity, which depends on the target word.

Therefore, we use a semi-supervised clustering approach for this task. In this approach, a constraint on a pair of data, set by the user, is used for clustering (Cohn et al., 2003)(Basu et al., 2004)(Bilenko et al., 2004)(Klein et al., 2002). The system selects some pairs from the data set, and offers them to a user. The user enters the pair's constraint (must-link or cannot-link) into the system. Must-link indicates that the two data items must belong to the same cluster, while cannot-link indicates that they cannot belong to the same cluster. The cost to provide such constraints is lower than that of the cost to assign a class label to each data.

2. Division of example sentences by semi-supervised clustering

2.1. Algorithm

Figure 1 shows our system algorithm.

First, the user enters a target word w, and the rough number k of meanings of w [1]. The system then collects the sentences, including w, from a corpus. The set of these sentences is the data set D. Next, the system divides D into k clusters with a general clustering tool.

$$C = \{C_1, C_2, \cdots, C_k\}$$

The set A in Figure 1 is the final clustering result, i.e., a set of clusters. The system first sets $A = \{C_1\}$, then sequentially picks up C_i from C, and evaluates whether the C_i must be added into A, or merged into a cluster in A. This is based on the user feedback. The system shows the central sentence of C_i and the central sentence of a cluster of C_j in A to the user. The user judges whether the meanings of the target words w in each sentence are identical. If they are identical, the user enters "must-link," and is merged into the cluster. If not ("cannot-link"), the procedure is repeated

[1] It is about from five to ten times of the estimated number.

for the next cluster of A. If the user judgments are "not identical" for every central sentence of A, the C_i is added into A. Note that the central sentence does not change on merging the two clusters, i.e., one central sentence is chosen and then continuously used. Therefore, the system output is unique, and the maximum number of user judgments is $k(k-1)/2$.

Input w and initial cluster number k
construct $\boldsymbol{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_N\}$
cluster D to k clusters $\boldsymbol{C} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \cdots, \boldsymbol{C}_k\}$
$\boldsymbol{A} = \{\boldsymbol{C}_1\}$, $\boldsymbol{C} = \{\boldsymbol{C}_2, \boldsymbol{C}_3, \cdots, \boldsymbol{C}_k\}$
for(i =2; i < k+1; i++) {
    z is the center of $\boldsymbol{C}_i$
    foreach C in A {
        x is the center of C
        give (x,z) to user
        get Const from user
        if (Const = must-link) {
            $\boldsymbol{C} \leftarrow \boldsymbol{C} \cup \boldsymbol{C}_i$; break
        }
    }
    if (Const = cannot-link) {
        $\boldsymbol{A} \leftarrow \boldsymbol{A} \cup \{\boldsymbol{C}_i\}$
    }
}
Output A

Figure 1: System algorithm

2.2. System description

Figure 2 $\sim$ 7 show our system. The system is implemented using Perl CGI. First, we input the target word (Figure 2).

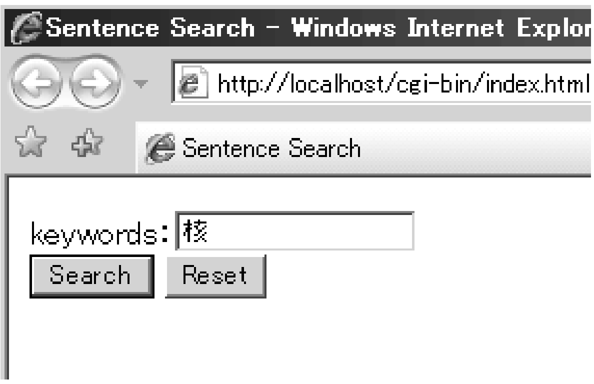

Figure 2: Input of a target word

The system retrieves example sentences including the target word from a corpus (Figure 3). As the search engine software, we use HyperEstraier provided in the following site.

```
http://hyperestraier.sourceforge.net/
  index.html
```

Figure 3: Example sentences

Then, the system conducts clustering for example sentences (Figure 4). As the clustering engine software, we use CLUTO provided in the following site.

```
http://glaros.dtc.umn.edu/gkhome/
  views/cluto
```

Figure 4: Initial clustering result

The number of clusters is given by the user. Then, the system selects a typical sentence from each cluster (Figure 5).

Figure 5: Typical sentences

The next stage is the semi-supervised clustering. The system shows a pair of typical sentences to the user, and the user judges whether target words in two sentences are used in same meaning, or not (Figure 6).

Figure 6: User judgment

After some iteration of this procedure, the system presents the final clustering result (Figure 7).

Figure 7: Final clustering result

2.3. Similarity between sentences

A measure of the similarity of the sentences is essential in our system.

The system converts a sentence into a feature list. Using this feature list, the system measures similarities between sentences. Our system uses the following features used in the paper (Shinnou and Sasaki, 2003). Suppose the target word is $w = w_i$, which is the i-th word in the sentence.

e1: the word w_{i-1}
e2: the word w_{i+1}
e3: two content words in front of w_i
e4: two content words behind w_i
e5: thesaurus ID number of e3 and e4

For example, let's consider the following sentence[2] in which the target word is *'kiroku(記録)'* [3].

kako/saikou/wo/kiroku/suru/ta/.

The system generates the following feature list from the above example sentence.

```
{ e1=wo, e2=suru, e3=saikou, e3=kako,
  e4=suru, e4=.,  e5=3192, e5=31920,
  e5=1164, e5=11642 }
```

Because of space limitations, we sketch out the similarity measurement $sim(A, B)$ between two feature lists, A and B.

The $sim(A, B)$ is the sum of three kinds of similarities.

```
sim(A,B) = 1/3 * {
 (Similarity of the feature e1)
 + (Similarity of the feature e2)
 + (Similarity of the feature e3,e4 and e5)}
```

The similarity is 1 if the feature has the same value, and 0 if it does not. In addition, the similarity is adjusted using some ad-hoc rules.

[2] A sentence is segmented into words, and each word is transformed into its original form by morphological analysis.

[3] The word 'kiroku(記録)' has at least two meanings: "memo" and "record".

2.4. Central sentence of cluster

We define the central sentence x_c of cluster

$$C = \{x_1, x_2, \cdots, x_n\}$$

as follows:

$$c = \arg\max_{i \in 1:n} \sum_{j \in 1:n} sim(x_i, x_j).$$

This indicates that the average of the similarities between x_c and sentences in C is the largest.

3. Experiment

We tested our system for 12 nouns (shown in Table 1) used in the SENSEVAL2 Japanese dictionary task (Shirai, 2003). The data set was constructed from the training and test data provided by SENSEVAL2. It gives 50 nouns, from which we picked only the words that produced 300 or more example sentences were picked. This procedure gave 12 nouns, which were then set as target words.

Table 1: Data sets

word	# of example sentences	# of meanings
mono(もの)	754	10
mondai(問題)	636	4
daihyou(代表)	466	3
mae(前)	426	4
kankei(関係)	414	3
gogo(午後)	396	3
jibun(自分)	362	2
jidai(時代)	360	4
kodomo(子供)	354	2
genzai(現在)	341	2
syakai(社会)	339	6
ima(今)	329	4
average	431.42	3.91

First, the user must enter the initial cluster number to the system by overestimating the number of possible meanings of the target word. In this experiment, the number was fixed at 20.

The system then converts a sentence into a feature list, and constructs the similarity matrix using the similarity measure, $sim(A, B)$. To perform clustering for the similarity matrix, we used the clustering tool kit CLUTO. We performed clustering using CLUTO with the cluster number set to 20, and without any optional parameters[4].

Next, through the user feedback described in the previous section, the system generates the final clustering result from the above 20 clusters. Table 2 shows the result.

The "# of clusters" in Table 2 lists the number of clusters generated by our system. The number in parentheses is the true number given by SENSEVAL2 using a dictionary. The "# of constraints" lists the number of constraints entered by the user. The "semi-supervised" shows the system accuracy, and "un-supervised" shows only the clustering accuracy.

Table 2: Result of experiment

target word	# of clusters	# of constrains	semi-super -vised	unsuper -vised
mono	4 (10)	66	0.391	0.309
mondai	1 (4)	19	0.969	0.969
daihyou	3 (3)	35	0.858	0.667
mae	3 (4)	24	0.855	0.371
kankei	2 (3)	30	0.785	0.848
gogo	3 (3)	27	0.634	0.444
jibun	1 (2)	22	0.942	0.942
jidai	2 (4)	28	0.653	0.550
kodomo	2 (2)	26	0.588	0.480
genzai	2 (2)	26	0.974	0.707
syakai	4 (6)	22	0.755	0.395
ima	3 (4)	23	0.687	0.423
Average	3.25 (3.91)	24.6	0.757	0.592

Table 2 shows that the performance using the semi-supervised approach is better than when using the unsupervised, and the number of constraints is small. Therefore, the system is efficient.

4. Discussion

4.1. Initial number of clusters

In our system, the user must provide the initial number of clusters for the target word; while in the experiment, the number was fixed at 20.

We varied the number of clusters from 10 to 100, in steps of 5. The result is shown in Figure 8. In Figure 8, the x-axis is the initial number of clusters entered by the user, the y-axis in the upper figure of Figure 8 is the average of 12 accuracies, and the y-axis in the lower portion of Figure 8 is the average number of constraints entered by the user.

It can be concluded, that the accuracy is higher when the number of initial clusters is large. However, a large number of clusters requires more constraints.

The initial number of clusters depends on the target word. If the target word is estimated to have many meanings, the initial number should be large. If not, we should set the initial number small. One of the advantages of our system is that the initial number of clusters is not fixed, and is instead entered by the user.

Accuracy primarily depends on the measurement of similarity. We plan to develop methods to further improve the accuracy of these measurements. We will need to construct a thesaurus for this task..

4.2. Similarity measurement

Precsion of our system depends on similarity measurement between example sentences. Our defined similarity measure is ad-hoc, so is needed to be improved. At the present, our system handles only noun words. For verb words, we must use case slots and syntactic information to measure similarity.

[4]The program used is "scluster," and its input is a similarity matrix.

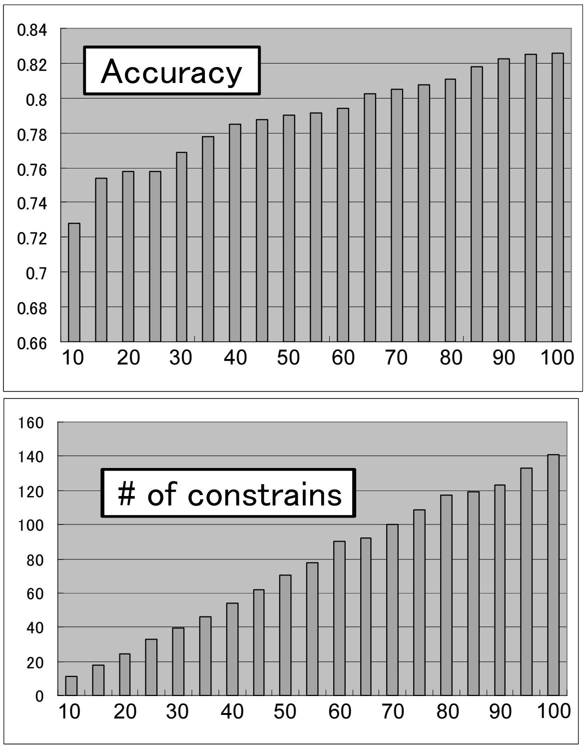

Figure 8: Constrains and accuracy for the initial cluster numbers

To measure similarity, use of a thesaurus is essential. We use Bunrui-goi-hyou[5] as the thesaurus. We can improve similarity measurement by using more powerful thesaurus. Improvement of similarity measurement is our future work. To do it, we must construct the thesaurus suitable for our task.

5. Conclusion

In this paper, we have described a system that divides example sentences based on the meaning of the target word. In this task, the estimation of the number of possible meanings is essential. Therefore, our system uses semi-supervised clustering. First, a data set is divided into many small clusters by using an initial clusters number entered by the user. Next, the system merges these clusters by asking the user for feedback. An experiment using 12 nouns demonstrated the high accuracy of our system, using a small number of constraints given by the user. In the future, we will improve the accuracy of the similarity measurements to obtain more accurate clustering results.

Acknowledgements

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research on Priority Areas, “ Japanese Corpus ”, 19011001, 2007.

[5]Japanese standard thesaurus

6. References

Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. 2004. Active semi-supervision for pairwise constrained clustering. pages 333–344.

Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. 2004. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In *ICML-2004*, pages 81–88.

David Cohn, Rich Caruana, and Andrew McCallum. 2003. Semi-supervised Clustering with User Feedback. Technical Report TR2003–1892, Cornell University.

Donald Hindle. 1990. Noun classification from predicate argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics (ACL-90)*, pages 268–275.

Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. 2002. From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering. In *ICML-2002*, pages 307–314.

Masaki Murata and Masao Utiyama and Kiyotaka Uchimoto and Qing Ma and Hitoshi Isahara. 2001. Japanese word sense disambiguation using the simple Bayes and support vector machine methods. In *Proceedings of the SENSEVAL-2*, pages 135–138.

Resnik Philip. 1992. WordNet and Distributional Analysis: A Class-based Approach to Lexical Discovery. In *Proceedings of AAAI-92 Workshop on Statistically-Based NLP Techniques*, pages 48–56.

Hiroyuki Shinnou and Minoru Sasaki. 2003. Unsupervised learning of word sense disambiguation rules by estimating an optimum iteration number in the EM algorithm. In *Proceedings of Seventh Conference on Natural Language Learning (CoNLL-2003)*, pages 41–48.

Kiyoaki Shirai. 2003. SENSEVAL-2 Japanese Dictionary Task (in Japanese). *Journal of Natural Language Processing*, 10(3):3–24.