Word Sense Disambiguation with a Corpus-based Semantic Network

Qujiang Peng Takeshi Ito Teiji Furugori

Department of Computer Science
University of Electro-Communications
1-5-1, Chofugaoka, Chofu, Tokyo 1828585, JAPAN
peng@phaeton.cs.uec.ac.jp, {ito, furugori}@cs.uec.ac.jp

Abstract

Determining the meaning of words in text is an important process in natural language processing. In this paper, we propose a new method for word sense disambiguation that uses a corpus-based semantic network. Creating a semantic network that represents semantic distances among words in general, we resolve the ambiguities activating the network. Theoretically, our method needs no annotation on the corpus from which a CSN is created and also makes the data sparseness problem irrelevant. Practically, it achieved a success rate of 92.1%, which is better than those of other comparable studies.

1. Introduction

Word sense disambiguation (WSD) has been a concern ever since the beginning days of computer treatment of natural language in the 1950s. The task is not an end in itself, but rather a necessary step at one level or another to have a better or complete system for natural language processing. We need to know for instance that the word *sentence* means a kind of punishment rather than a group of words in translating *Taro got a heavy sentence for the crime he committed* into any language.

We had developed a WSD system [11]. It was corpus- and similarity- based like the one by Dagan and others [6]. Unlike theirs, however, we considered not the local context but the global context of polysemous words, and tried to lessen the data sparseness problem, for which they gave no thought, by using WordNet [18], a lexical database for English.

The system attempted to disambiguate 682 instances of 10 polysemous words and got a success rate of 91.5%. Its result was impressive compared to other studies [1), 5), 7)].

In this paper we offer yet another WSD system. We base this work on a corpus-based semantic network (CSN) and try to improve the performance further, as well as eliminating some deficiencies observed in our previous experiment.

2. Background

The majority of studies done recently in WSD use some kinds of corpora and statistical measures to determine the meaning of words in sentences. As are well-known, Yarowsky presented a statistical method for resolving lexical ambiguities with the use of Roget's Thesaurus and a large corpus [16]. A 100-word context of each member in Roget's Thesaurus category was extracted from the corpus, and a mutual-information-like estimate was made to identify words most likely to co-occur with the category members. He used the words thus found to disambiguate the meanings for new occurrences of a polysemous word. Dagan and Itai proposed a method for WSD in one language using statistical data from a monolingual corpus of another language. They chose the preferred sense according to a statistical model on lexical relations in the target language put in a constraint propagation algorithm [5]. Karov and Edelman devised a method using a text corpus and machine readable dictionary [9]. Their system learns from the corpus a set of typical usages for each of the senses of a polysemous word listed in the dictionary and assigns the sense associated with the

typical usage most similar to its context to the new instance of a polysemous word.

The idea of using semantic networks in natural language processing is nothing new [12]. Many have used them in WSD, too [8), 14), 15)]. Hiro, Wu and Furugori [8], for instance, attempted to disambiguate the meanings of polysemous words using a CSN of 1,691 nodes built from LOB corpus, a corpus of 1,006,815 words, in which each node (word) has a hundred branches or links to other nodes. Figure 1 shows a portion of the network.

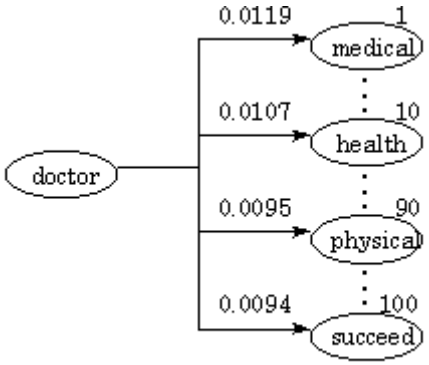

Figure 1. A portion of semantic network

The natural number i in Figure 1 indicates ith word ($1 \le i \le 100$) that has ith highest association to the word *doctor*. The real number shows the strength of association between the word *doctor* and the word in ith node.

Activating the network, they tried to determine the meaning of a polysemous word in the following four steps:

(a) Get the context vector, C, of 1,691 elements by the words in the portion of text containing a polysemous word w.
(b) Get the context vectors for w's senses, $S_1, S_2, \cdots, S_k$, of 1,691 elements each by the words contained in the definitions of w in a dictionary.
(c) Calculate the Euclidean distance between C and S_i ($1 \le i \le k$).
(d) Select the sense associated with S_i whose distance to C is the smallest as the meaning of the polysemous word.

3. Lexical Disambiguation with a Corpus-based Semantic Network

3.1 Construction of Corpus-based Semantic Network

We employ a CSN to that of Hiro, Wu, and Furugori [8]. Our CSN is built from EDR corpus [17], a corpus of 160,000 sentences, and consists of 1,845 nodes, each having a hundred links to others. It contains all the content words (nouns, verbs, adjectives) whose frequency counts in EDR corpus are bigger than 60.

We calculate the association value between the words w_1 and w_2 in the network using mutual information [4]:

$$I(w_1, w_2) = \log_2 \left(\frac{N_1 * f(w_1, w_2)}{f(w_1) f(w_2)} \right) \quad (1)$$

Here, N_1 in (1) is the size of the corpus used in the estimation, $f(w_1, w_2)$ is the frequency of co-occurrences of w_1 and w_2, and $f(w_1)$ and $f(w_2)$ is the frequency of each word. If there is a strong association between w_1 and w_2, then $I(w_1, w_2) >> 0$. If there is a weak association between w_1 and w_2, then $I(w_1, w_2) \approx 0$. If $I(w_1, w_2) << 0$, then w_1 and w_2 are said to be in complementary distribution.

The link from node n_i to n_j in the CSN has the weight given by the value of the mutual information for the words w_i and w_j.

3.2 Processes of Word Sense Disambiguation

A CSN built from a corpus is "colored" by the domain the corpus deals with. We use such a network to determine the meaning of a polysemous word appearing in text.

Procedure Before the disambiguation processes start, we collect from the

textual data on the Internet the sentences that contain the polysemous words to be disambiguated. We then classify the sentences for each polysemous word by its senses manually and select six instances each. Finally, we determine the meaning of a polysemous word using this data, a text in which the polysemous word appears, and the CSN. Its steps are:

(a) Activate the CSN using the six instances and get the node vector, N_L, of certain length L for each sense of a polysemous word w.
(b) Activate the CSN using a portion of the text in which w appears and get the context vector, V_L, of certain length L, and its corresponding node vector.
(c) Calculate the similarity, using V_L, between each of the node vectors in (a) and the node vector in (b).
(d) Select the sense that got the maximal similarity value as the meaning of w in the text.

Let us see first how we spread the activation over the CSN and how we define and use the vectors.

Activation of CSN Activation is spread when a word w_k in an instance or in the portion of the text for the polysemous word w matches the word for the node n_k in the network ($w_k = n_k$). We use the equation (2) to calculate the potential, a_i, of the node n_i $(i = 1,2,\cdots,1845)$ at time $t+1$.

$$a_i(t+1) = \begin{cases} a_i(t) + I(n_i, n_k) & \textit{if } n_i \textit{ and } n_k \textit{ have link in the CSN} \\ a_i(t) & \textit{otherwise} \end{cases} \quad (2)$$

Context vector and node vector The input from the text at time $t-1$ spreads over the network at time t and produce an activation pattern, $P(t)$.

$$P(t) = (a_1(t), a_2(t), \cdots, a_{1845}(t)) \quad (3)$$

$P(t)$ gives an influence to $P(t+1)$ and it in turn to $P(t+2)$, and so on.

The pattern $P(t+l)$ from time t to $t+l$ using a window of l words each before and after w is:

$$P(t+l) = (a_1(t+l), a_2(t+l), \cdots, a_{1845}(t+l)) \quad (4)$$

The vector N of corresponding nodes to this pattern is:

$$N = (n_1, n_2, \cdots, n_{1845}) \quad (5)$$

We get $P'(t+l)$ by arranging the elements in (4) in decreasing order.

$$P'(t+l) = (a'_1(t+l), a'_2(t+l), \cdots, a'_L(t+l), \cdots, a'_{1845}(t+l)) \quad (6)$$

We see in (6) that $a'_1(t+l) \geq a'_2(t+l) \geq \cdots \geq a'_L(t+l) \geq \cdots \geq a'_{1845}(t+l)$. The vector N' of corresponding nodes to this pattern is:

$$N' = (n'_1, n'_2, \cdots, n'_L, \cdots, n'_{1845}) \quad (7)$$

The relevant nodes to express the domain represented in the context of w come in front part of N' as the number of activating the CSN increases. We call this vector the context vector V_L and its corresponding node vector the node vector N_L.

$$V_L = \phi(a'_1(t+l), a'_2(t+l), \cdots, a'_L(t+l)) = (b_1, b_2, \cdots, b_L) \quad (8)$$

$$N_L = (n'_1, n'_2, \cdots, n'_L) \quad (9)$$

Here, $1 \le L \le 1845$ and ϕ is the normalization factor that restricts the value of b_i to $[0, 1]$.

Determination of the meaning Miller and Charles [10] found evidence in several experiments that humans determine the semantic similarity of words from the similarity of the contexts the words are used in. Extending the finding, Schütze [13] hypothesized that the same holds for word senses: senses are interpreted as groups (or clusters) of similar contexts of the ambiguous word. Karov and Edelman [9] used the idea in their study of WSD that words are considered similar if they appear in similar contexts and contexts are similar if they contain similar words.

These in mind, let us see how we calculate the similarity is measured.

Let s_m be the m th sense of a polysemous word w. Using the instances from the Internet with $l = 50$, we get the node vector N_L of s_m:

$$N_L(s_m) = (x_1, x_2, \cdots, x_L)$$

Similarly, we get the context vector V_L and the node vector N_L from the context representation (CR) of 50 words each before and after a polysemous word w in text:

$$V_L(CR) = (y_1, y_2, \cdots, y_L)$$
$$N_L(CR) = (z_1, z_2, \cdots, z_L).$$

From these, we calculate the similarity $Sim(CR, s_m)$ in the equation:

$$Sim(CR, s_m) = \sum_{i=1}^{L} \sum_{j=1}^{L} y_j \times A(x_i, z_j) \quad (10)$$

Here,

$$A(x_i, z_j) = \begin{cases} 1 & if \quad x_i = z_j \\ 0 & otherwise \end{cases} \quad (11)$$

In (10), we first get the set $N_L(s_m) \cap N_L(CR)$, and then calculate its ratio in $N_L(CR)$ by the context vector $V_L(CR)$.

4. Experiment and Rusult

We tested our method for the same 10 polysemous words used in our previous work with 756 instances collected randomly from [19), 20), 21)] and other materials. The processes of getting the meaning of w are:

(a) Obtain the $N_L(s_1)$, $N_L(s_2), \cdots$, $N_L(s_r)$ for the lexical meaning of w.

(b) Get $V_L(CR)$ and $N_L(CR)$ using a window of l words each before and after w in text.

(c) Calculate the similarity $Sim(CR, s_1)$, $Sim(CR, s_2), \cdots$, $Sim(CR, s_r)$.

(d) Select s_m that got the maximal similarity value to be the meaning of w.

Table 1 shows the 10 polysemous words, their senses, the number of instances, and the instances identified correctly in our experiment using $L = 400$.

4.1 Illustrated Examples

We illustrate how the processes work in an example. Suppose the word tested is *palm* in the following text:

> *CR* = ⋯ them suddenly from slumber. Beside each sleeper lay his weapon--these were never far from their owners from childhood to death. The sight of the swords made the young man's palm itch. He stepped quickly to them, selecting two short swords--one for Kar Komak, the other for himself; also some trappings for his naked comrade. Then he started ⋯ (*a text on Internet*)

Palm is given two nominal meanings s_1 and s_2: *tree* and *hand*. We generate the context vector and node vectors:

$N_L(tree)$= (*tree, vegetable, flower, animal, fish, fruit, river, garden, farm,* ⋯)

$N_L(hand)$=(*left, weight, leg, shoulder, doctor, arm, mother, pull, eye, injury,* ⋯)

Table 1: Polysemous Words Tested

Words	Senses	Instances	Resolved(%)
band	group of musicians	19	18 (94.7)
	strip or stripe	12	11 (91.7)
cabinet	administrative organ	24	23 (95.8)
	shelf	17	16 (94.1)
court	judicial	163	153(93.9)
	area for ball game	19	18 (94.7)
crane	machine	16	14 (87.5)
	bird	21	20 (95.2)
palm	tree	20	19 (95.0)
	hand	52	49 (94.2)
plant	living thing	86	79 (91.9)
	factory	25	19 (76.0)
sentence	group of words	41	36 (87.8)
	punishment	67	61 (91.0)
slug	bullet	26	24 (92.3)
	animal	16	14 (87.5)
tank	combat vehicle	13	11 (84.6)
	water-filled place	13	13 (100)
trial	action of judging	89	82 (92.1)
	test	17	16 (94.1)

$N_L(CR)$= (*ban, abolish, square, boy, strict, gather, track, occupy, arm, negotiate,* ⋯)

$V_L(CR)$=(*0.016169, 0.015220, 0.014355, 0.014296, 0.013498, 0.013326, 0.013245, 0.012949, 0.012911, 0.012909,* ⋯)

The similarities calculated for $Sim(CR, s_1)$ and $Sim(CR, s_2)$ are 0.062193 and 0.135044 with $L = 100$. We thus get the meaning of *palm* to be s_2 (*hand*) and which is correct in this instance.

Take one more example for the word *trial* in:

> *CR* = ⋯ courts of probates (analogous in certain matters to the spiritual courts in England), a court of admiralty and a court of chancery. In the courts of common law only, the trial by jury prevails, and this with some exceptions. In all the others a single judge presides, and proceeds in general either according to the course of the canon or civil law, without the aid of ⋯ (*Also a text on Internet*)

Again, *trial* is given two nominal meanings s_1 and s_2 : *action of judging* and *test*. The context vector and node vectors are:

$N_L(s_1)$= (*criminal, court, trial, judge, prison, lawyer, guilty, violate, appeal,* ⋯)

$N_L(s_2)$= (*test, virus, skill, blood, medical, examination, compile, signal, data,* ⋯)

$N_L(CR)$=(*ruling, district, suit, hearing, criminal, lawyer, prosecutor, trial, violate, battle,* ⋯)

$V_L(CR)$=(*0.016464, 0.016406, 0.016110, 0.015863, 0.015717, 0.015336, 0.014982, 0.014866, 0.014854, 0.014722,* ⋯)

The similarities $Sim(CR, s_1)$ and $Sim(CR, s_2)$ are 0.500731 and 0.020939 for $L = 100$. So we get the meaning of the *trial* to be s_1.

4.2 Results

Table 2 shows the disambiguation results (success rates) for L = 25, 50, 100, 200, 300, 400, 600, 800, 1000, 1200, 1400, 1600, and 1845, respectively.

Table 2: Disambiguation Results (%)

L \ Words	band	cabinet	court	crane	palm	plant	sentence	slug	tank	trial	Average
25	90.3	90.2	90.1	89.2	91.7	83.8	85.2	81.0	96.2	86.8	87.8
50	93.5	95.1	87.9	89.2	90.3	87.4	86.1	83.3	96.2	86.8	88.4
100	93.5	92.7	91.8	89.2	95.8	90.1	88.9	85.7	92.3	90.6	91.0
200	93.5	92.7	94.0	89.2	94.4	89.2	89.8	90.5	92.3	91.5	91.8
300	93.5	92.7	93.4	89.2	94.4	88.3	88.0	90.5	96.2	93.4	91.7
400	93.5	95.1	94.0	91.9	94.4	88.3	89.8	90.5	92.3	92.5	92.1
600	93.5	95.1	94.0	86.5	95.8	89.2	89.8	90.5	92.3	92.5	92.1
800	93.5	95.1	94.0	86.5	88.9	89.2	91.7	81.0	100	88.7	90.9
1000	93.5	95.1	92.3	89.2	90.3	86.5	89.8	88.1	96.2	90.6	90.6
1200	93.5	95.1	90.7	89.2	97.2	86.5	89.8	90.5	96.2	91.5	91.0
1400	93.5	95.1	92.3	91.9	93.1	78.4	88.9	88.1	92.3	90.6	89.4
1600	90.3	85.4	90.1	89.2	84.7	69.4	70.4	88.1	92.3	89.6	83.3
1845	90.3	82.9	81.9	89.2	90.3	67.6	57.4	83.3	73.1	92.5	79.1

As is seen in Table 2, the best result is 92.1% for L = 400 or 600, which is far better than that of Hiro, Wu and Furugori and better than many other experiments [2), 6), 16)].

Figure 2 shows the average results for the various values of L from 25 to 1845.

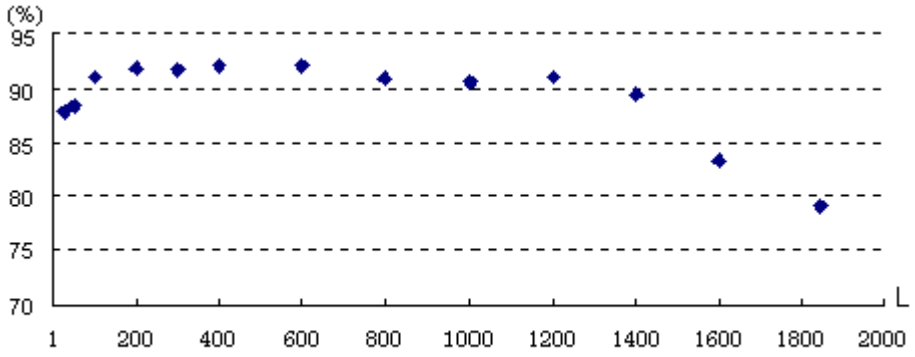

Figure 2. Result of the disambiguation experiment

The result is not very good when L is too small or too big. Naturally, the disambiguating information to express the domain involved is not sufficient enough when L is too small and the vector contains too much noise information when L is too big.

4.3 Evaluation

We presented an approach for WSD based on a statistical measure of word similarities in the previous work. In it, we first obtained contextual-similarity vectors for the senses of a polysemous word, making use of a tagged corpus, and defined also the contextual representation for the polysemous word appearing in text. We then calculated distributional matrix between each contextual-similarity vector and the contextual representation for the word to be disambiguated. Finally, comparing the density values of distributional matrices, we selected the sense with the highest value as the meaning of the polysemous word.

In the current model based on a CSN, we achieved little better success rate than 91.5% of the previous work. But it offers two distinct methodological advantages: we need no tagged corpus and made the data sparseness problem irrelevant.

The upper-bound of the success rate is supposed to be 100%, which is that of human beings against which we measured the success rates. The lower-bound or the base-line success rate is 74.1%, which is the sum of bigger instances in each word divided by the total instances, i.e., 560/756. This number is nothing to compare with our result of 92.1% in Table 2.

Our system sometimes fails, however. For instance, we got a wrong result for *plant* in:

> ··· An immaterial but visible being that inhabited the air when the air was an element and before it was fatally polluted with plant smoke, sewer gas and similar products of civilization. Sylphs were allied to gnomes, nymphs and salamanders, which dwelt, respectively, in earth, water and fire, ···(*Also a text on Internet*)

There are two possible reasons why we got *living thing* rather than *factory* in this example. The one is that we had relevant words more to *living thing* than to *factory*, resulting more information for the former in $V_L(CR)$ and $N_L(CR)$. The other is that the words related to *living thing* are general enough so that they are represented in the CSN but the words related to *factory* sense of *plant* are more specific and less represented in the CSN. If so, such a failure may be eliminated if we create the CSN from a corpus that is tuned more to the domain the polysemous words to be disambiguated are used in.

Many used the definition of a polysemous word in a dictionary when calculating the $N_L(s_m)$ [3), 8)]. We found that the definitions of a word are often too short and too uneven to cover necessary collocations, resulting in not enough information to activate the CSN or in the activation of irrelevant portions of the CSN. We avoided this happening by the use of instances from the Internet text in the calculation of the $N_L(s_m)$.

5. Conclusion

We proposed a new method for word sense disambiguation based on a semantic network as the means for determining the sense of polysemous words. This work is distinct as well as advantageous over many other studies in the following points.

Generally,

- It needs no annotation on the corpus from which we create a CSN.
- It made the data sparseness problem irrelevant.

In more specific accounts,

- The equation in (2) for activating the CSN is more concise and produces less noises on the CSN compared to the one by Hiro, Wu, and Furugori [8)].
- Using L, we get better vectors, capturing effective information, for

calculating the $Sim(CR, s_m)$ than the ones by many other work [8), 13)].

. The calculation of $Sim(CR, s_m)$ in the equation (10) is simpler and more intuitive than the one by Hiro, Wu, and Furugori [8)].

As noted in section 4, the performance of our system is consistent and better than others. We are sure that it will be improved further if we build the CSN from a corpus in an area such as economics or medicine and test polysemous words in that area because the "color" of the CSN becomes more distinct. The use of EDR corpus was not ideal one in this respect.

References

1) Black, E. An Experiment in Computational Discrimination of English Word Senses. *IBM Journal of Research and Development*, Vol.32, No.2, pp.185-194 (1988).

2) Chen, J. N., and Chang, J. S. A Concept-based Adaptive Approach to Word Sense Disambiguation. In *Proceedings of COLING-ACL '98*, Montreal, Quebec, Canada, pp. 237-243, (1998).

3) Chen, J.N., and Chang, J. S. Topical Clustering of MRD Senses Based on Information Retrieval Techniques. *Computational Linguistics*, Vol.24, No.1, pp.61-95 (1998).

4) Church, K., and Hanks, P. Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics*, Vol.16, pp.22-29 (1990).

5) Dagan, I., and Itai, A. Word Sense Disambiguation Using a Second Language Monolingual Corpus. *Computational Linguistics*, Vol.20, No.4, pp.563-595 (1994).

6) Dagan, I., Marcus, S., and Markovitch, S. Contextual Word Similarity and Estimation from Sparse Data. *Computer Speech and Language*, Vol.9, pp.123-152 (1995).

7) Hearst, M. Noun Homograph Disambiguation Using Local Context in Large Text Corpora. *Using Corpora*, University of Waterloo, Waterloo, Ontario (1991).

8) Hiro, K., Wu, H., and Furugori, T. Word-Sense Disambiguation with a Corpus-Based Semantic Network. *Journal of Quantitative Linguistics*, Vol.3, pp.244-251 (1996).

9) Karov, Y., and Edelman, S. Similarity-based Word Sense Disambiguation. *Computational Linguistics*, Vol.24, No.1, pp.41-59 (1998).

10) Miller, G. A., and Walter G. Charles. Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes*, Vol.6, No.1, pp.1-28 (1991).

11) Peng, Q., Takakura, S., and Furugori, T. Determination of the Meaning of Polysemous Words Using a Word Similarity Measurement. *SIG Notes, NL-142, Information Processing Society of Japan*, No. 20, pp.59-66 (2001).

12) Quillian, M. R.: Semantic Memory, In Minsky, M. (ed.): *Semantic Information Processing*, MIT Press, pp.227-266 (1968).

13) Schütze, H. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1), pp.97-123 (1998).

14) Towell, G., and Voorhees, E. M. Disambiguating Highly Ambiguous Words. *Computational Linguistics*, Vol.24, No.1, pp.125-145 (1998).

15) Veronis, J., and Ide, N. M. Word Sense Disambiguation with very Large Neural Networks Extracted from Machine Readable Dictionaries. In *Proceedings of COLING-90*, Helsinki: ICCL, pp.389-394 (1990).

16) Yarowsky, D. Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of COLING-92*, Nantes:ICCL, pp. 454-460 (1992).

17)http://www.iijnet.or.jp/edr/index.html

18)http://www.cogsci.princeton.edu/~wn

19)http://www.usnews.com/usnews/

20)http://gamp.c.u-tokyo.ac.jp/archive/textdb.htm

21)http://www.infomotions.com/etexts/