Word Sense Disambiguation with Information Retrieval Technique

Jong-Hoon Oh, Saim Shin, Yong-Seok Choi, and Key-Sun Choi

Division of Computer Science, Dept. of EECS, Korea Advanced Institute of
Science and Technology 373-1 Guseong-dong Yuseong-gu Deajeon 305-701 Korea
{rovellia, miror, angelove, kschoi}@world.kaist.ac.kr

Abstract

This paper reports on word sense disambiguation of Korean nouns with information retrieval technique. First, context vectors are constructed using contextual words in training data. Then, the words in the context vector are weighted with local density. Each sense of a target word is represented as 'Static Sense Vector' in word space, which is the centroid of the context vectors. Contextual noise is removed using selective sampling. A selective sampling method use information retrieval technique, so as to enhance the discriminative power. We regard training samples as indexed documents and test samples as queries. We can retrieve relevant top-N training samples for a query (a test sample) and construct 'Dynamic Sense Vector' using the retrieved training samples. A word sense is estimated using the 'Static Sense Vector' and 'Dynamic Sense Vector'. The Korean SENSEVAL test suit is used for this experiment and our method produces relatively good results.

1. Introduction

Word sense disambiguation is a potentially crucial work in many NLP applications such as machine translation (Brown *et al*., 1991), parsing (Lytinen, 1986), and information retrieval (Krovets *et al*., 1992; Voorhees 1993). There have been many studies on corpus-based word sense disambiguation (WSD) (Agirre *et al*., 1996; Esscudero *et al*., 2000; Gale *et al*., 1992; Gruber, 1991; Hinrich, 1998). They mainly use the words in limited window-sized context. Co-occurring words within a limited window-sized context are used as clues for supporting one sense among the semantically ambiguous ones. Our method is also a corpus-based approach. The problem is to find the most effective patterns in training data to capture the right sense. It is true that they have similar context when words are used with the same sense (Rigau *et al*., 1997). However, if training samples contain noise, it is difficult to capture effective patterns for WSD (Atsushi *et al*., 1998). To filter out the noise, we use a selective sampling method. A selective sampling method uses information retrieval technique, so as to enhance the discriminative power. If there are training samples and a test sample, we can select training samples, which contextual words are similar to those of the test sample. The selected training samples may have more discriminative powers because there are more contextual words corresponding to those in a test sample than others. To select the training samples, we use information retrieval technique. Training samples are regarded as indexed documents and test samples are regarded as queries. Then we can retrieve relevant top-N training samples for a query (a test sample).

We also use another feature – local density. If contextual words frequently co-occur with certain sense of target nouns, they may be strong evidence to support the sense. Moreover, it is true that words nearby an ambiguous word give more effective patterns or features than those far from it (Jen *et al*., 1998). Words in context are weighted with local density, which is based on distance and relative frequency of the contextual words.

This paper is organized as follows: section 2 shows the methods we applied. Section 3 deals with experiments, and section 4 discusses the errors. Conclusion and future works are drawn in sections 5.

2. Method

2.1. System Description

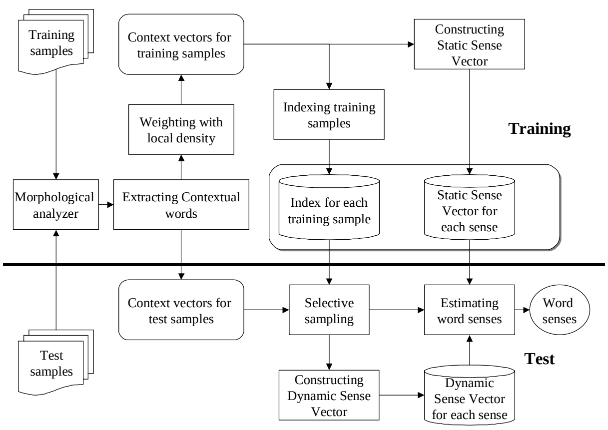

Figure 1: overall system description

Figure 1 shows the overall system description. The system is composed of a training phase and a test phase. In the training phase, contextual words in the limited context window are extracted from training samples (Extracting Contextual words). Then the contextual words are weighted by their distance from a target noun and their distribution in the training samples of each sense (Weighting with local density). Each training sample can be represented as context vectors with its contextual words. Now, we can construct a sense vector called 'Static Sense Vector' by clustering context vectors of training samples for each sense (Constructing Static Sense Vector). 'Static Sense Vector' is the centroid of context vectors of all training samples for each sense. Let contextual words of a training sample for a target noun 'bank' (sense1: financial institution, sense2: shore) be 'business', 'commercial', and 'money' for sense1 and be 'fish', 'river' and 'water' for sense2. If there are two context vectors for sense1 – ('business', 'money') and ('business', 'commercial') – and two context vectors for sense2 –

('fish', 'river') and ('river', 'water') –, we can acquire 'Static Sense Vector' for sense1 ('business: weight', 'commercial: weight', 'money: weight') and for sense2 ('fish: weight', 'river: weight', 'water: weight'). Next, we index each training sample with its contextual words for a selective sampling method (Indexing training samples).

In the test phase, contextual words are extracted with the same manner as in the training phase. Then, we select relevant training samples for a given test sample so as to capture effective patterns for WSD (Selective sampling). In this paper, this procedure is called as 'selective sampling'. The selective sampling module selects N training samples using the cosine similarity between indexed training samples and the contextual words of a given test sample. We can make another sense vectors for each sense with the selected training samples (Constructing Dynamic Sense Vector). Since, the sense vectors produced in the selective sampling procedure are changed according to the contextual words in a test sample, we call the sense vector 'Dynamic Sense Vector' in this paper. 'Dynamic Sense Vector' is also the centroid of context vectors for each sense. Contrary to 'Static Sense Vector', 'Dynamic Sense Vector' is constructed by clustering not all training samples but selected training samples for each sense.

Finally, word sense for target nouns are estimated using 'Static Sense Vector' and 'Dynamic Sense Vector' (Estimating word senses). The cosine similarities between the two kinds of sense vector and context vectors of a test sample make it possible to estimate a word sense. The sense with the highest similarity is selected as the right word sense.

2.2. Representing Training Samples as Context Vectors using Local Density

The window size of context is fixed to five sentences including one sentence for the target noun. Context must reflect various contextual characteristics[1]. If the window size of context is too large, the context cannot contain relevant information consistently (Kilgarriff, 2000). Words in the context window are classified into nouns, verbs, and adjectives. The classified words within the context window are assumed to show the co-occurring behaviour with the target noun. They give the supporting vector weighted by their relative frequencies for senses and their distance from the target noun. Modifiers of a target noun help word sense disambiguation. For example, *bam*[2] has two senses: 'night' and 'chestnut'. In the context "delicious *bam*", the sense of '*bam*' tends to be 'chestnut' rather than 'night'. On the other hand, "dark *bam*" is to be "dark night" rather than "dark chestnut". Words nearby a target noun give more information to decide its sense than those far from it. Distance from a target noun is used for this purpose. It is calculated by the assumption that target nouns in the same context have the same sense (Yarowsky, 1995).

Each word in the training samples can be weighted by formula (1). Let $W_{ij}(t_k)$ represent a weighting function of a term t_k, which appears in the j^{th} training sample for the i^{th} sense, tf_{ijk} represent the frequency of the term t_k in the j^{th} training sample for the i^{th} sense, df_{ik} represent the number of training samples for the i^{th} sense where the term t_k appears, D_{ijk} represent the average distance of t_k from the target noun in the j^{th} training sample for the i^{th} sense, and N_i represent the number of training samples for the i^{th} sense, which contain a term t_k.

$$W_{ij}(t_k) = \frac{1}{Z} \times \left(\log(tf_{ijk}+1) \times \frac{1}{\sqrt{D_{ijk}}} \times \frac{df_{ik}}{DF_k} \times \frac{N}{N_i} \right) \quad (1)$$

$$N = \sum_{i}^{\#\_of\_senses} N_i, \quad DF_k = \sum_{i}^{\#\_of\_senses} df_{ik}$$

$$Z = \sqrt{\sum_{k=1}^{\#\_of\_term} \left(\log(tf_{ijk}+1) \times \frac{1}{\sqrt{D_{ijk}}} \times \frac{df_{ik}}{DF_k} \times \frac{N}{N_i} \right)^2}$$

In formula (1), Z is a normalization factor, which forces all values of $W_{ij}(t_k)$ to fall into between 0 and 1, inclusive. Formula (1) is variation of tf-idf. In tf-idf, tf means the Term Frequency, the number of times a particular term occurs in a given document and idf means the Inverse Document Frequency, a measure of how often a particular term appears across all of the documents in a collection. They are typically used for weighting the parameters of a model. Tf-idf is a popular method for weighting terms in the information retrieval domain (Salton *et al*., 1983).

D_{ijk}, df_{ik}, and N_i in formula (1) support a local density concept. Local density in this paper means not only word distance from a target noun but also relative frequency of contextual words. If contextual words co-occur with certain sense of target nouns frequently, they may be strong evidence to support the sense. With the local density concept, context of training samples can be represented by a vector with context words and their weight, such that $(W_{ij}(t_1), W_{ij}(t_2), \ldots W_{ij}(t_n))$. When $W_{ij}(t_k)$ is 1, it means that t_k is strong evidence for the i^{th} sense.

2.3. Constructing Static Sense Vectors

We represented training samples as vectors in the previous section. Now, we can represent each sense of a target noun as sense vectors. A sense vector for certain sense can be acquired by clustering context vectors of training samples, which contain a target noun having the sense. Since context vectors are in the vector space which axis is contextual words, we calculate the centroid of context vectors for each sense to acquire the sense vector. Because the sense vectors are not changed according to test samples, we call them 'Static Sense Vector' in this paper (note that 'Dynamic Sense Vector', which we will describe in section 2.4, is changed according to context of test samples).

Let v_{ij} be the context vector of the j^{th} training sample for the i^{th} sense, and N_i be the number of training samples for the i^{th} sense. The 'Static Sense Vector' for the i^{th} sense,

[1] POS, collocations, semantic word associations, sub-categorization information, semantic roles, selectional preferences and frequency of senses are useful for disambiguating an occurrence of a word (Agirre 2001)

[2] Korean romanised transcription will be written in the italic script.

SV_i, is represented by formula (2). In formula (2), SV_i is the centroid of context vectors for the i^{th} sense as shown Figure 2 (Park, 1997).

In Figure 2, there are n senses and context vectors, which represent each training sample. We can categorize each context vector according to a sense of a target word. Then, each sense vectors is acquired using formula (2).

$$SV_i = \frac{\sum_{j=1}^{|N_i|} v_{ij}}{|N_i|} \quad (2)$$

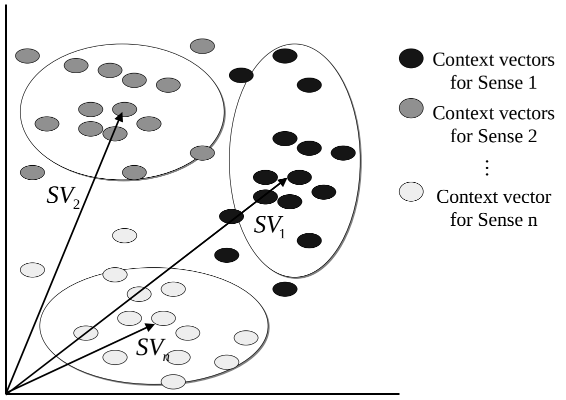

Figure 2: Graphical representation of 'Static Sense Vector'

2.4. Selective Sampling: Dynamic Sense Vectors

It is important to capture effective patterns and features from training data in WSD. If there is noise in the training data, it makes difficult to disambiguate word senses effectively. To reduce negative effects of the noise, we use a selective sampling method using information retrieval technique. Figure 3 shows the process of a selective sampling method. There are n senses for a target noun and indexed training samples for each sense. For the given context vector of a test sample – we regard it as a query –, top-N training samples can be retrieved by cosine similarity (Salton *et. al*., 1983). Because we know a target word in training samples and test samples, we can restrict search space into training samples, which contain the target word when we find relevant samples. The retrieved samples for each sense are used for constructing a sense vector for each sense.

Consider the case that there are n different queries for information retrieval system. Then the retrieved results will be different. The same situation is occurred in our selective sampling method. For n different context vectors of test samples, the selective sampling method will retrieve different top-N training samples. Therefore, the sense vector produced in this step is dynamically changed according to a context vector of a test sample. This is the reason why we call it 'Dynamic Sense Vector' in this paper.

Let RT_i be the number of retrieved training samples for the i^{th} sense in the top-N, and v_{ij} be the context vector of the j^{th} training sample for the i^{th} sense in the top-N. The 'Dynamic Sense Vector' for the i^{th} sense of a target noun, DSV_i, is formulated by formula (3). In formula (3), DSV_i means the centroid of context vectors of retrieved training samples for the i^{th} sense as shown in the lower side of Figure 3.

$$DSV_i = \frac{\sum_{j=1}^{|RT_i|} v_{ij}}{|RT_i|} \quad (3)$$

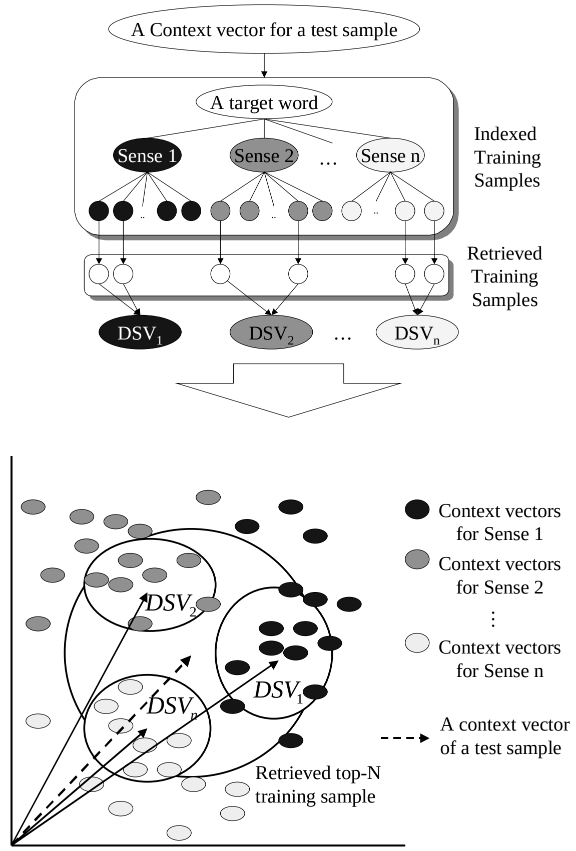

Figure 3: Graphical representation of a selective sampling method using information retrieval technique: the upper side shows retrieval process for the context vector of a test sample and the lower side shows graphical representation of 'Dynamic Sense Vector' for each sense

2.5. Context Vectors of Test Data

Contextual words in a test sample are extracted as the same manner in the training phase. The classified words in the limited window size – nouns, verbs, and adjectives – offer components of context vectors. When a term t_k appears in the test sample, the value of t_k in a context vector of the test sample will be 1, in contrary, when t_k does not appear in the test sample, the value of t_k in a context vector of the test sample will be 0. Let contextual words of a test sample be 'bank', 'river' and 'water', and dimension of a context vector be ('bank', 'commercial', 'money', 'river', 'water'). Then we can acquire a context vector, CV =(1,0,0,1,1), from the test sample. Henceforth we will denote CV_i as a context vector for the i^{th} test sample.

2.6. Comparing Similarities

We described the method for constructing 'Static Sense Vector', 'Dynamic Sense Vector' and context vectors of a test sample. Next, we will describe the method for estimating a word sense using them. The similarity in information retrieval area is the measure of how alike two documents are, or how alike a document and a query are. In a vector space model, this is usually interpreted as how close their corresponding vector representations are to each other. A popular method is to compute the cosine of the angle between the vectors (Salton *et al.*, 1983). Since our method is based on a vector space model, the cosine measure (formula (4)) will be used as the similarity measure.

Throughout comparing similarity between SV_i and CV_i and between DSV_i and CV_i for the i^{th} sense and the j^{th} test sample, we can estimate the relevant word sense for the given context vector of the test sample. Formula (5) shows a combining method of $sim(SV_i,CV_j)$ and $sim(DSV_i,CV_j)$. Let CV_j represent the context vector of the j^{th} test sample, s_i represent the i^{th} sense of a target word, and $Score(s_i,CV_j)$ represent score between s_i and CV_j.

$$sim(v,w) = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2 \sum_{i=1}^{N} w_i^2}} \quad (4)$$

where, N represents the dimension of the vector space, v and w represent vectors.

$$\arg\max_{s_i} Score(s_i, CV_j) = \lambda \times sim(SV_i, CV_j) + (1-\lambda) \times sim(DSV_i, CV_j) \quad (5)$$

where, λ is a weighting parameter.

Because the value of cosine similarity falls into between 0 and 1, that of $Score(s_i,CV_j)$ also exists between 0 and 1. When the similarity value is 1, it means perfect consensus, in contrary, when the similarity value is 0, it means that there is no part of agreement at all. After all, the sense having maximum similarity by formula (5) is decided as the right word sense.

3. Experiment

3.1. Experimental Setup

In this paper, we evaluate six systems as follows.

- The system that assigns a word sense, which appears most frequently in the training samples (Baseline)
- The system with the Naïve Bayesian method (A) (Gale *et al.*, 1992)
- The system with only 'Static Sense Vector' weighted by word frequencies (B)
- The system with only 'Static Sense Vector' weighted by local density ($\lambda = 1$) (C)
- The system with only 'Dynamic Sense Vector' ($\lambda = 0$) (D)
- The system by the proposed method (with Top N= 50, $\lambda = 0.2$, the value of N and λ is determined by cross validation) (E)

Word	Training sample	Test sample	Baseline
'*mal*'	118	34	23.53%
'*noon*'	133	66	95.45%
'*son*'	132	66	95.45%
'*baram*'	101	50	90.00%
'*geoli*'	234	67	62.69%
'*jail*'	98	52	67.31%
'*euisa*'	160	85	71.76%
'*mok*'	98	50	96.00%
'*jeom*'	106	42	80.95%
'*bam*'	97	53	81.13%

Table 1: Distribution of the test suit

The test suit is the Korean lexical samples released for SENSEVAL-2 in 2001. This test suit supplies training data and test data for 10 nouns (SENSEVAL-2, 2001). Senses of each noun are described in appendix. Table 1 shows the number of training samples and test samples for each word.

Cross-validation on training data is used to determine the parameters – λ in formula (5) and *top-N* in constructing 'Dynamic Sense Vector'. We divide training data into ten folds with the equal size, and determine each parameter, which makes the best result in average from ten-fold validation. The values, we used, are $\lambda = 0.2$, and *N=50*.

The results were evaluated by precision rates (Salton *et al.*, 1983). The precision rate is defined as the proportion of the correct answers to the generated results.

3.2. Experimental Result

Table 2 shows the performance of each system. This results show that they have different precisions by their processing and training methods although they use the same training data. In system B and C, we find that local density gives more discriminative powers to 'Static Sense Vector'. Results of C and D show that 'Dynamic Sense Vector' is useful for WSD. This indicates that reducing noise and selecting relevant sample give more effective sense vectors for WSD.

In the result, there are some cases where 'Static Sense Vector' is effective and some cases where 'Dynamic Sense Vector' is effective. By getting their strong points, the proposed method (E) shows higher performance. Our method also shows higher performance than that of the Naïve Bayesian method (Gale *et al*., 1992).

As a result of this experiment, we proved that context information throughout local density and selective sampling is more suitable and discriminative in WSD. This techniques lead up to about 84.5% performance improvement in the experiment comparing the system A (Naïve Bayesian). We also show that combination of 'Static Sense Vector' and 'Dynamic Sense Vector' makes better results.

Local density improves the performance about 54% (between 'B' and 'C') and selective sampling shows improvement about 7.52% (between 'C' and 'D').

Word	Base-line	A	B	C	D	E
'mal'	23.53%	26.47%	20.59%	32.35%	23.53%	41.18%
'noon'	95.45%	7.58%	77.27%	95.45%	96.97%	96.97%
'son'	95.45%	12.12%	4.55%	84.85%	96.97%	96.97%
'baram'	90.00%	22.00%	40.00%	88.00%	92.00%	96.00%
'geoli'	62.69%	58.21%	64.18%	40.30%	67.16%	79.10%
'jali'	67.31%	26.92%	17.31%	71.15%	76.92%	76.92%
'euisa'	71.76%	85.88%	61.18%	81.18%	90.59%	90.59%
'mok'	96.00%	62.00%	50.00%	96.00%	98.00%	98.00%
'jeom'	80.95%	71.43%	73.81%	80.95%	80.95%	80.95%
'bam'	81.13%	84.91%	83.02%	94.34%	84.91%	86.79%
Total	78.23%	46.90%	50.44%	77.70%	83.54%	86.55%

Table 2: Experimental results

4. Analyzing Errors

We analyzed errors after experiment. The errors are classified into two main causes –by Korean morphemes and by insufficiency of training data.

Errors caused by Korean morphemes can be classified two types. One is the ambiguity of Korean morphemes and the other is productivity of Korean nouns. In the experiment, '*gin* (long)', which is wrongly analyzed, makes difficult to disambiguate the word sense 'street' of '*geoli*'. '*ginja*' frequently appears in the sample, where '*geoli*' is used as the sense, 'street'. However, '*ginja*', which meaning is the name of the street in Tokyo, is wrongly analyzed as '*gin* (long) + *ja* (ruler)'. Because '*gin* (long)' mainly supports another sense, 'distance', of '*geoli*', the '*gin*' has a negative effect when a test sample contains '*ginja*' and the correct sense of '*geoli*' is 'street'.

One spacing unit in Korean is called a word phrase. A typical word phrase consists of a sequence of content words (like noun or verb stem) and functional words (like postposition or verbal ending). In Korean, a compound noun can be in a word phrase. Sometimes, this may cause errors because the compound noun can be segmented into several nouns. For example, '*bolissal* (polished barley)' in a word phrase is combination of '*boli* (barley)' and '*ssal* (rice)' in Korean. Consider the case that there is a noun '*boli*', which is strong evidence for certain sense in training sample and there is '*bolissal*' but '*boli*' in test sample. If '*bolissal*' is not analysed as not '*boli* (noun)'+ '*ssal* (noun)' but '*bolissal* (noun)', it does not offer the strong evidence for determining the sense. Moreover, a verb derived from a noun makes the same problem. In Korean some nouns can be extended to verbs just by attaching an affix '*~ha*'. For example, a noun '*mal* (language)' can be extended to a verb '*malha* (speak)'. The verb derived from a noun can be analyzed as a verb itself or a noun and an affix. It makes the same problem as that of the base noun in a compound noun. It will be necessary to handle the property to reduce the problems.

Our experimental data is not large size and this test suit was extracted from various documents. If sense distribution of certain word in the training data has preponderance that is a common phenomenon in raw corpus, the sparse senses show lower precision than other senses because of insufficiency of training data, such as '*mal*', and '*jeom*'.

5. Conclusion and Future works

This paper reports about word sense disambiguation in Korean nouns. Our method is summarized as follows.

- Training Phases
  1. Constructing context vectors using contextual words in training data.
  2. Local density to weight contextual words in context vectors.
  3. Creating 'Static Sense Vector', which is the centroid of the context vectors of the whole training data.
- Test Phases
  1. Constructing context vectors using contextual words in test data.
  2. Selective sampling for each test case to reduce noise.
  3. Creating 'Dynamic Sense Vector', which is the centroid of the selectively sampled training data for each sense.
  4. Estimating word senses using static and dynamic sense vectors.

Our method improves performance about 7.5% ~ 84.5% precision in the experiment comparing the system without local density and selective sampling.

Our method is somewhat language independent, because it needs only POS information. If there is a morphological analyzer for one language, our method can disambiguate ambiguous senses of words for the language. We will apply our method to other languages such as English. Though our method produces relatively good results, there are scopes to improve the performance. In analyzing errors, we find that the productivity of Korean nouns and the ambiguity of Korean morpheme is one of the main reasons of errors. In future work, we will show their effects on Korean WSD. Because we just use information in a morphological level, there are scopes to improve the performance by using additional information in a syntactic and a semantic level – dependency relations, approximated word senses of context words, and collocations are possible (Agirre, 2001).

6. Reference

Agirre E. and Rigau G. (1996). Word Sense Disambiguation using Conceptual Density, *In Proceedings of 16th International Conference on Computational Linguistics, COLING'96*. Copenhagen, Denmark.

Agirre E. and Martinez D. (2001). Knowledge Sources for Word Sense Disambiguation. *In Proceedings of the Fourth International Conference TSD 2001*.

Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. (1998). Selective Sampling for Example-based Word Sense Disambiguation. *Computational Linguistics*, 24(4), pp. 573-597

Brown, Peter F., Stephen A. Della Pietra, and Vincent J. Della Pietra. (1991). Word-sense disambiguation using statistical methods. *In Proceedings of the 29th Annual Meeting*, 264-270.

Escudero G., Màrquez L. and Rigau G. (2000). Boosting Applied to Word Sense Disambiguation. *In Proceedings of the 11th European Conference on Machine Learning, ECML 2000. Barcelona, Spain. Lecture Notes in Artificial Intelligence 1810*. R. L. de Mántaras and E. Plaza (Eds.). Springer Verlag,

Gale, William A., Kenneth W. Church, and David Yarowsky. (1992). A method for disambiguating word senses in a large corpus. *Computers and Humanities*, 26, 415-439

Gruber T. R. (1991). Subject-Dependent Co-occurrence and Word Sense Disambiguation. *In Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*

Hinrich Schutze. (1998). Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1), 97-123

Jen Nan Chen and Jason S. Chang. (1998). A Concept-based Adaptive Approach to Word Sense Disambiguation. *In Proceedings of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING/ACL-98*, pp 237-243

Kilgarriff A. and J. Rosenzweig. (2000). English SENSEVAL: Report and Results. *In Proceedings of 2nd International Conference on Language Resources & Evaluation LREC' 2000*, Athens.

Krovets, Robert and W. Bruce Croft. (1992). Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2), 115-141

Lytinen, Steven L. (1986). Dynamically combining syntax and semantics in natural language processing. *In Proceedings of AAAI-86*, 574-578

Park.Y.C. (1997). *Building word knowledge for information retrieval using statistical information.* Ph.D. thesis, Department of Computer Science, Korea Advanced Institute of Science and Technology

Rigau G., Atserias J. and Agirre E. (1997). Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. *In Proceedings of joint 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics ACL/EACL'97*. Madrid, Spain.

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*, New-York: McGraw-Hill

SENSEVAL-2. (2001). http://www.sle.sharp.co.uk/senseval2/

Voorhees, Ellen M. (1993). Using WordNet to disambiguate word senses for text retrieval. *In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and development in Information retrieval*, 171-180

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivalling Supervised Methods. *In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, MA, pp. 189-196

Appendix

Word	Sense	Sense id	Word	Sense	Sense id
‘*mal*’	The soldier in stick games like chess.	k00001	‘*jail*’	Space or seat	k00051
	End	k00002		Position	k00052
	Horse	k00003		Opportunity	k00053
	The unit of cereals or liquid.	k00004		A figure in number	k00054
	Language	k00005	‘*euisa*’	Doctor	k00061
‘*noon*’	Eye	k00011		Pretending to die.	k00062
	The part connected knot with another knot like net.	k00012		Death for justice	k00063
	Snow	k00013		Mind	k00064
‘*son*’	Hand	k00021		Pseudo word.	k00065
	Younger people	k00022		Deliberation	k00066
	Damage	k00023		Official rank in ‘*Shinra*’.	k00067
	Helping	k00024		Medicine	k00068
	Descendant	k00025		Similar to the real	k00069
	Visitor	k00026	‘*mok*’	Neck	k00071
	Power of one’s own.	k00027		The similar part whose shape is similar to neck.	k00072
‘*baram*’	Wind	k00031		Important and narrow place that can’t go out without it like pathway.	k00073
	Hope	k00032		Tree	k00074
	Mode about something.	k00033	‘*jeom*’	Dot, spot	k00081
	One’s appearance or conduct without the necessary.	k00034		Point of view.	k00082
‘*geoli*’	Street or road.	k00041		An item.	k00083
	Material or data to do something like cooking.	k00042		A piece.	k00084
	A large profit.	k00044	‘*bam*’	A Chestnut	k00091
	Act or scene in drama.	k00045		Night	k00092
	Distance	k00046			

Table: The sense dictionary of a target noun