Unsupervised Learning for Verb Sense Disambiguation Using Both Trigger Words and Parsing Relations*

Cheng Niu, Zhaohui Zheng , Rohini Srihari, Huifeng Li, Wei Li

Cymfony Inc., 600 Essjay Road, Williamsville, NY 14221, USA
{cniu, zzheng, rohini, hli, wei}@Cymfony.com

This paper presents an unsupervised machine learning approach to verb sense disambiguation based on context clustering. The context is represented as a combination of co-occurring trigger words within a window size of the verb and the parsing relations of the verb. The context is retrieved from a large indexed repository that stores a raw corpus of 88,000,000 words and its corresponding parsing results. The clusters are mapped onto the senses defined by the SENEVAL-2 standards. Benchmarking shows that when using trigger words only, this approach produces results reaching state-of-the-art performance in the unsupervised category. When parsing relations are combined with triggers words in integrated training, a modest performance enhancement is observed. Alternative approaches in using the parsing relations for contextual clustering are discussed.

Key words: Word Sense Disambiguation, Unsupervised Learning, Context Clustering

1 INTRODUCTION

Word Sense Disambiguation (WSD) is one of the central problems in Natural Language Processing. Word senses are highly subject to domain effects: it is often the case that a sense used in one domain may be invalid in another domain. Therefore, domain portability is the key for the usability of a WSD system in supporting natural language applications. The unsupervised learning approach demonstrates its domain portability advantages in overcoming the serious 'knowledge bottleneck' faced by the hand-crafted rule approach [Kelly & Stone 1975; Hirst 1987] and supervised learning approach [Gale *et al* 1992; Brown *et al* 1991].

Traditionally, WSD research uses an external thesaurus (mainly WordNet) to pre-define the target senses. Recently, there is significant progress in automatic thesaurus construction using a corpus-driven approach. [Lin 1998; Lin 2002].

Among the major categories, Noun, Adjective and Verb, verb sense disambiguation (VSD) is the most difficult task.[1] However, VSD is of particular value since verb is the axis of the sentential argument structure. When a verb is properly disambiguated in its context, the major content of a sentence is captured. This paper focuses on VSD, but the presented techniques should be equally applicable to other categories.

Most WSD systems either use selectional restriction in parsing relations, or use trigger words that co-occur within a window size of the ambiguous word. Our machine learning approach attempts to combine both.

The unsupervised learning approach to VSD presented in this paper is based on *context clustering*. The context is represented as a combination of co-occurring trigger words and the re-

* This work was partly supported by the Navy SBIR program under contract N00178-02-C-3073.

[1] Based on results reported in SENSEVAL-2, the scores for verbs are usually 7~10% lower than noun and adjective [http://www.sle.sharp.co.uk/senseval2/Results/guidelines.htm#rawdata].

lated parsing relations. The context is retrieved from a large indexed repository. The clusters are mapped onto the senses defined by the SENEVAL2 standards. Benchmarking shows: (i) when using only trigger words, this approach produces results comparable with state-of-the-art performance in the unsupervised category; (ii) when parsing relations are combined with triggers words in integrated training, modest performance enhancement is observed.

2 RELATED WORK

There has been significant research on WSD using unsupervised learning. [Yarowsky 1995] presents an approach using bootstrapped machine learning from a raw corpus. This approach requires predefined word senses and uses trigger words only. [Lin 1997; Resnik 1997] present unsupervised WSD systems which rely exclusively on parsing relations.

[Schutze 1998] presents a context clustering approach. A word sense is represented as the context cluster in which the word is used in that sense. Our research adopts this concept, however, there are a number of differences:

(i) Instead of using only trigger words, the context in our definition is represented as a combination of trigger words and parsing relations.
(ii) We have explored automatic estimation of the number of senses.
(iii) Our work involves a sense mapping process to a thesaurus used in the SENSEVAL2 community. As a result, our VSD system can be directly compared with other systems using the community standards.

3 SYSTEM DESIGN AND ALGORITHM

We have implemented the two-step approach as follows: (i) *Sense discrimination*: represent context containing the target verb as a combination of trigger words and parsing relations for clustering, the resulting context clusters correspond to the senses of the verb; (ii) *Sense mapping*: map the context clusters onto an external thesaurus by comparing the clusters with the example sentences contained in the thesaurus.

3.1 System Design

Figure 1 shows the system architecture for the VSD training

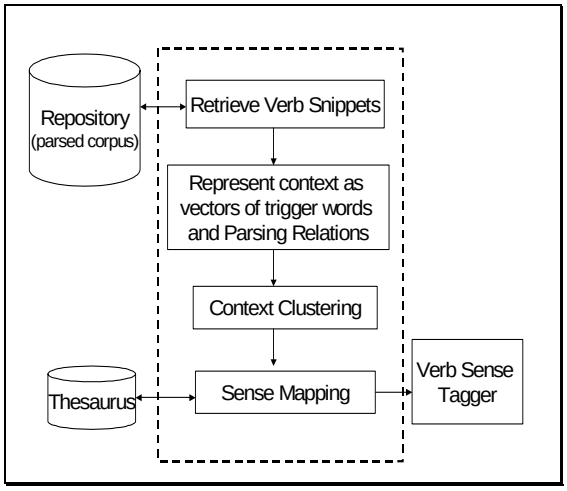

Figure 1. System Architecture for VSD Training

Before unsupervised learning is started, a large raw corpus of ~8,8000,000 words (1.2 GB) is parsed. The parsed corpus is saved into Repository, which supports fast retrieval by a keyword based indexing scheme. The training consists of four steps:

1. Given a verb, retrieve all the snippets containing the verb: a snippet is defined as a token sequence in a window;
2. Select informative trigger words and parsing relations, and represent context as vectors of the trigger words and parsing relations;
3. Cluster the contexts using *k*-means [Rasmussen 1992];
4. Map clusters onto senses by classifying example sentences associated with each sense from an external thesaurus.

For each verb, a sense tagger is generated from this training process. The learned sense taggers disambiguate verbs by comparing the incoming context of a verb with existing context clusters associated with the verb.

3.2 VSD Using Only Trigger Words

The VSD algorithm using only trigger words is adopted from [Schutze 1998] with modification in feature selection and context representation.

Trigger words are those words whose occurrences are significantly associated with the occurrence of the verb. There are two commonly used approaches in measuring the significance of association: χ^2 and uncertainty coefficient. Both are used in our research. Equation 1 defines χ^2 [Press *et al* 1993]:

Equation 1. $$\chi^2 = \sum_{i,j=0,1} \frac{(N_{ij} - n_{ij})^2}{n_{ij}}$$

where N_{00} denotes the number of sentences that contain both the verb and the trigger word; N_{01} denotes the number of sentences that contain the verb but not the trigger word; N_{10} denotes the number of sentences that contain the trigger word but not the verb, and N_{11} denotes the number of sentences that contain neither the verb nor the trigger word. n_{ij} is the expectation value of N_{ij} based on null hypothesis (Equation 2):

Equation 2. $$n_{ij} = \frac{N_i^0 N_j^1}{N}$$

where N_0^0 (N_0^1) denotes the number of sentences containing the verb (trigger word) while N_1^0 (N_1^1) denotes the number of sentences not containing the verb (trigger word). N denotes the total number of sentences in the repository.

Uncertainty coefficient is defined in Equation 3 [Press *et al* 1993]:

Equation 3. $$U(tw, verb) = \frac{MI(tw, verb)}{-\sum_i \frac{N_i^1}{N} \log \frac{N_i^1}{N}}$$

where $MI(tw, verb)$ denotes the mutual information between the trigger word and the verb, and is defined in Equation 4.

Equation 4. $$MI(tw, verb) = -\sum_{i,j} \frac{N_{ij}}{N} \log \frac{N_{ij} N}{N_i^0 N_j^1}$$

In which N_{ij}, N_i^k and N refer to the same quantities used for χ^2 computation.

We use the χ^2 measure to select the trigger words and use the uncertainty coefficient to characterize the strength of the association [Press *et al* 1993].

In our experiment, the trigger words are first sorted based on χ^2, and the top 1,000 trigger words are selected. Then, the uncertainty coefficient of each selected trigger word is computed.

Table 1 shows a sample of the top trigger words for the verb *save*:

Table 1: Trigger word list for verb *save*

Trigger word	Uncertainty coefficient
Money	0.039
Life	0.0232
Time	0.0100
Cost	0.00806
Energy	0.0045
Million	0.0038
retirement	0.00297
Annually	0.00261
Reduce	0.00201
goalkeeper	0.00181

To derive verb senses, the snippets that contain the verb are retrieved from the indexed repository. In this experiment, the range of a snippet is defined as 50 tokens to the left and 50 tokens to the right of the verb inspired by the empirical results shown in [Gale *et al* 1992].

For commonly used verbs, a large number of snippets are available in the repository. For example, more than 20,000 snippets are retrieved for the verb *run*. Since some snippets contain few informative trigger words that can contribute to the identification of word senses, removing such snippets will make the resulting clusters more distinct and will improve the clustering performance. A ranking module is implemented to filter out the less informative snippets. Snippets are sorted based on the summation of the mutual information of all the trigger words in the snippet. Only the top *n* snippets are used for clustering. In this experiment, $n = 2{,}000$.

In the process of clustering, context is represented as a vector, and each entry of the vector corresponds to a trigger word. The value of each vector entry depends on the presence of the trigger words in the snippet. The length of the vector is normalized to one.

Equation 5.
$$v = \{v_0, v_1, v_2 \cdots v_n\} \frac{1}{\sqrt{\sum_i v_i^2}}$$

$$v_i = \begin{cases} \mathrm{U}(w_i, verb), \text{ if } \textit{i-th} \text{ trigger word is present} \\ 0, \text{otherwise} \end{cases}$$

The dissimilarity between the context vectors is defined as the standard Euclidian distance. The *K*-means algorithm [Rasmussen 1992] is used for context clustering. Standard *K*-means is the most commonly used clustering algorithm employing Euclidian distance in machine learning. However, it may get stuck in a local minimum when the starting points are not properly selected. In this paper, we address this problem by combining the *Maximin* clustering algorithm with *K*-means, referred to as modified *K*-means. We first apply Maximin clustering algorithm to find *K*

"good" starting points and then do the standard *K*-means clustering based on those points. The procedure of the modified *K*-means is as follows.

1. Choose an arbitrary data point from the data set as the first initial point.
2. Choose as the next candidate the data point that has the greatest distance to the set of initial points selected so far. Note: the distance between a point and a set is defined as the minimum distance between the point and each point in that set.
3. Check whether the number of initial points selected so far is *K* and if Yes, go to Step 4; otherwise, go to Step 2.
4. Do *K*-means clustering using the above *K* initial points.

Finally, we combine the results of the standard and modified *K*-means approaches to get better clustering performance by selecting the one with minimum within-cluster dissimilarity out of several runs over the two approaches.

In the clustering process, context vectors associated with the same document are forced into the same cluster, following the *one sense per discourse* principle [Gale *et al* 1992].

To facilitate the checking of the clustering results, clusters can be presented using their most distinctive trigger words. The distinctive trigger words are the trigger words that have a strong tendency to occur in a specific cluster. The distinctive trigger words are selected based on χ^2 between the trigger words and a cluster. The resulting clusters for *serve* are shown below. The number of clusters is predefined as 5.

cluster 1:
World War II, veteran, Army, country, member, years, Navy, WWII, retire, Korean War, U.S. Army, Vietnam, military, dinner, U.S Navy, meal, Korea, war, officer, sauce

cluster 2:
board, chairman, United Way, Moose International, Missouri Bankers Association, Swalec, Nazarene Youth International council, Aurora Easter Seals, House of Delegates, FSA Group, Copley Memorial Hospital, trustee, governor, association, president, committee, vice chair, ADA, appoint, executive vice president

cluster 3:
customer, business, wireless, network, market, company, bank, utility, provider, residential, metropolitan area, worldwide, large, ebusiness, provide, consumer, electric, effectively, financial

cluster 4:
sentence, prison, life, convict, parole, murder, jail, conviction, judge, Omar Abdel-Rahman, John Tobin, early release, American Fulbright, probation, federal, six months, Terry Nichols, Al-Sallam, Corey Shannon, Abdeen Jabara

cluster 5:
community, Diane Watson, California Senate, black woman, population, diocese, people, Mariana Patricia Jimenez, inner-city, perpetuate, Chiapas, profession, mentor, entire, diverse, public, school, plead, institution, volunteer

It is easy to make sense of the first four clusters, which correspond fairly well to four distinct senses of *serve*. The fifth cluster does not seem to be a distinctive cluster, and is probably constructed only to fit the pre-defined number of clusters. This motivates the research of dynamic sense number estimation (Section 3.4).

3.3 VSD Combining Trigger Words and Parsing Relations

To study potential VSD performance enhancement with parsing support, the following parsing relations are added to the feature space for clustering.

Four types of parsing relations, to be defined below, are used for the VSD task: *V_S*, *V_O*, *V_AdvM*, *V_PPM*. These are all directional dependency links from the verb to its directly linked dependant nodes. Note that our parser consumes surface structures and decodes the underlying logical relations. Both active patterns and passive patterns are parsed into same underlying dependency links.

(1) V_S: from verb to its logical subject
  e.g. *The algorithm designed by John works* →
  V_S(*design, John*)
  V_S(*work, algorithm*)
(2) V_O: from verb to its logical object
  e.g. *This company was founded to provide new telecommunication services* →
  V_O(*found, company*)
  V_O(*provide, service*)
(3) V_AdvM: from verb to its adverb modifier
  e.g. *He works diligently* →
  V_AdvM(*work, diligently*)
(4) V_PPM: from verb to its Prepositional Phrase modifier
  e.g. *He works for a small company* →
  V_PPM(*work, <for, company>*)

For each verb, not only the trigger words but also the parsing relations involving the verb are retrieved. Unlike the trigger word selection based on χ^2, the relations are selected based on their occurrence frequency due to the fact that parsing relations are much sparser than trigger words. Sparseness of relations makes it difficult to derive accurate evaluation of χ^2.

The uncertainty coefficient of a parsing relation $R(w, verb)$ can be computed using Equation 6.

Equation 6. $$U(R(w,verb),verb) = \frac{-\sum_{i,j}\frac{n_{ij}}{n}\log\frac{n_{ij}n}{n_i^0 n_j^1}}{-\sum_i \frac{n_i^1}{n}\log\frac{n_i^1}{n}}$$

$$n_{00} = \left|R(w,verb)\right|,$$

$$n_{01} = \sum_{w' \neq w}\left|R(w',verb)\right|,\ n_{10} = \sum_{verb' \neq verb}\left|R(w,verb')\right|, n_{11} = \sum_{\substack{w \neq w' \\ verb' \neq verb}}\left|R(w',verb')\right|$$

$$n_0^0 = \sum_{w'}\left|R(w',verb)\right|$$

$$n_1^0 = \sum_{w,verb'=verb}\left|R(w',verb')\right|$$

$$n_0^1 = \sum_{verb'}\left|R(w,verb')\right|$$

$$n_1^1 = \sum_{w' \neq w, verb'} \left| R(w', verb') \right|$$

$$n = \sum_{w', verb'} \left| R(w', verb') \right|,$$

where $\left| R(x, y) \right|$ denotes the occurrence count of the parsing relation $R(x, y)$.

The following are the most frequent parsing relations involving verb *save* and their uncertainty coefficients:

Table 2: Parsing relationship list for *save*

Parsing relationship	Uncertainty coefficient
V_O(*save, money*)	0.0227
V_O(*save, life*)	0.0285
V_O(*save, time*)	0.00433
V_O(*save, cost*)	0.00125
V_O(*save, energy*)	0.00843
V_O(*save, world*)	0.00201
V_O(*save, space*)	0.00222
V_S(*save, move*)	0.00519

The snippet is represented as a vector in the extended feature space that combines both trigger words and parsing relations. If a trigger word or a parsing relation is present, the corresponding entry will contain the value of the uncertainty coefficient. The length of vector is normalized to one.

The dissimilarity between context vectors is defined as Euclidian distance. The clustering process remains the same as for trigger words alone. The following are the clustering results for *serve* using both trigger words and parsing relations, also with predefined clustering number 5.

cluster 1:

V_PPM(*serve, <in, Army>*),*Army, World War II,* V_PPM(*serve, <during, World War II>*), *Vietnam War, retire, Mr., Air Force,* V_S(*serve, Myers*), *Wis,* V_S(*serve, captain*), V_O(*serve, Wis.*),V_PPM(*serve, <during, <Korean War>*), V_AdvM(*abroad*), *sunday school,* V_S(*serve, George*),V_PPM(*serve, <at, Pentagon>*), *Signal Corps*

cluster 2:

board, member, V_PPM(*serve, <on, board>*), *purpose, community,* V_O(*serve, community*), *country, president,* V_O(*serve, client*), *committee,* V_O(*serve, purpose*), *interest, include, client, organization, years,* V_O(*serve, interest*), *Fortune 500, elect, term*

cluster 3:

customer, V_O(*serve, customer*), *business,* V_AdvM(*serve, globally*), *provider,* V_O(*serve, carrier*), *utility, wireless, market, residential, worldwide, company,* V_S(*serve, Unisys*), *ebusiness, effectively, network, bank,* V_O(*serve, majority*), *large*

cluster 4:

V_O(*serve, sentence*), *sentence, life, convict, murder, prison, conviction, Omar Abdel-Rahman, John Tobin, jail, parole,* V_PPM(*serve, <on, charge>*), *American Fulbright, early release,* V_S(*serve, cleric*), *probation,* V_PPM(*serve, <for, conviction>*), V_PPM(*serve, <for, role>*), *Abdeen Jabara, Al-Sallam*

cluster 5:

dinner, V_O(*serve, dinner*), *flight, coleslaw*, V_S(*serve, dinner*), V_S(*serve, alcohol*), *lunch, salad, burgers, eat, personal check*, V_O(*serve, burgers*), *spicy*, V_O(*serve, menu*), *chef*, V_S(*serve, food*), *Café, breakfast, wine, southern*

Interestingly enough, the first four clusters for both methods are consistent with their respective four senses. The trigger-word-based clustering only uncovers the first four senses, and the last cluster is very noisy. However, the last cluster derived using both trigger words and parsing relations points to a fairly distinctive sense used in food service (e.g. *serve dinner*). This indicates that the evidence from the parsing relations helps to uncover an additional sense.

4 BENCHMARKING USING SENSEVAL-2

Corpus-driven snippet clusters need to map to a verb sense standard before the performance benchmark can be compared with those benchmarks of other systems using that same standard. We use the course grain lexical sample task in SENSEVAL-2 as benchmarking standard. The benchmarking procedure is described as follows.

i) Process the SENSEVAL-2 training corpus using our parser;
ii) For each verb to be benchmarked, cluster the related contexts as described in Section 3;
iii) Compute the centroids for all the contexts associated with a particular verb sense in the SENEVAL2 training corpus;
iv) Map the clusters to the SENSEVAL-2 verb senses by comparing the dissimilarity between the cluster centroids and the sense centroids: we allow for multiple clusters to be mapped onto one sense;
v) Parse SENSEVAL-2 testing corpus;
vi) Classify the context of each verb occurrence in the testing corpus into one of the clusters;
vii) Tag the verb with the sense corresponding to the cluster.

In this experiment, we set the number of clusters to 5 for all the tested verbs. Table 3 shows the accuracy benchmarking for each of the 29 verbs from the SENSEVAL-2 testing corpus. Our VSD performance is 50.6%, which is better than the top performer (44.53% for verbs) recorded in SENSEVAL-2 in the unsupervised WSD category.

Compared with trigger-word-based VSD, the combined model involving both trigger words and parsing relations shows modest performance enhancement, one percentage point. Our analysis shows that the slight performance enhancement is mainly due to the limited situations when the trigger word model alone fails but the learned parsing-based rules get applied. Usually, the words linked in the parsing relations are also trigger words and the cases where they drive towards different senses are not widespread to allow for a significant performance boom.

Our parsing relation evidence involves keyword-based selectional restriction which checks the specific word of a parsing relation. There are two limitations of the keyword-based selectional restriction. First, only a small portion of the VSD phenomena can be captured by the keyword-based selectional restriction used in our experiment. Such keyword-based selectional restriction has two problems.

(i) In the training stage, the associated phenomena are too sparse for effective VSD learning: in fact only the strongest collocational relationships such as ‘play…role’, ‘serve…purpose’, etc. may be captured; currently selectional restriction phenomena involving semantic classes such as ‘human’, ‘animate’, ‘ab-

stract', etc. cannot be captured. An example of class based selectional restriction is as follows, the sense of *work* can be disambiguated by checking if its subject is 'human' or not, ('John works hard' vs. 'This method works).'

(ii) In the tagging stage, the chances for the keyword-based selectional restriction rules to be fired are small.

Table 3: Benchmarking Using SENSEVAL-2

Verb	Num-ber	Top Performer Supervised	Top Performer Un-supervised	Our VSD using trigger words only	Our VSD using both trigger words and parsing relations.
Begin	280	87.5	29.8	53.2	53.6
Call	66	66.7	33.3	53.0	53.0
Carry	66	50.0	24.2	28.8	28.8
Collaborate	30	90.0	90.0	90.0	90.0
Develop	69	49.3	42.0	43.5	43.5
Draw	41	43.9	22.0	22.0	22.0
Dress	59	87.4	83.1	61.0	61.0
Drift	32	62.5	34.4	43.8	43.8
Drive	42	69.0	45.2	66.7	66.7
Face	93	90.3	71.5	90.3	90.3
Ferret	1	100	100	100	100
Find	68	39.7	17.2	33.8	35.2
Keep	67	46.3	25.4	34.3	37.3
Leave	66	53.0	50.0	34.8	31.8
Live	67	68.7	45.5	50.7	50.7
Match	42	59.5	50.0	69.0	64.2
Play	66	51.5	34.8	36.4	40.9
Pull	60	68.3	40.0	38.3	40.0
Replace	45	88.9	87.8	93.3	93.3
See	69	42.0	13.8	20.3	20.3
Serve	51	54.9	36.3	27.5	49.0
Strike	54	51.9	38.9	24.1	24.1
Train	63	52.4	26.5	39.7	39.7
Treat	44	79.5	54.5	45.5	45.5
Turn	67	53.7	44.8	28.4	28.4
Use	76	84.2	78.9	78.9	78.9
Wander	50	90.0	68.0	90.0	92.0
Wash	12	83.3	41.7	50.0	50.0
Work	60	58.3	51.7	46.7	46.7
Total	1806	66.91	**44.53**	**49.6**	**50.6**

Secondly, when the clustering tendency based on selectional restriction is not consistent with the one based on trigger words, the learned clusters may be blurred due to the integrated nature of using two diverse types of evidence. For example, the selectional restriction "leave…behind" may co-occur with several trigger-word-based clusters. This involves the issue of choosing between an integrated VSD system and a pipeline VSD system.

The modest performance enhancement brought by parsing relations does not indicate that the parsing evidence is not worth using or that the generally recognized selectional restriction is not effective for VSD. However it does show that keyword-based selectional restriction has its limitation and calls for more sophisticated learning algorithm involving class-based abstraction and a flexible system architecture that can maximize the benefits of the two diverse evidence categories.

.

5 CONCLUSION

We have explored an unsupervised WSD learning method for verbs using techniques of context clustering. The performance of the implemented system is among the best in the unsupervised category.

Our method adds keyword-based selectional restriction to trigger words in integrated learning. The contribution from the keyword-based selectional restriction is found to be modest for this method.

The analysis of the results indicate two directions for future research that may lead to more effective use of evidence from parsing relations: (i) extend the keyword-based selectional restriction to class-based selectional restriction to increase the coverage of the phenomena that can be captured by rules based on parsing relations; (ii) explore techniques that externally coordinate the trigger word-based rules and the parsing relation-based rules instead of the integrated training method.

REFERENCES

Brown, P.F., S.D. Pietra, V.J.D. Pietra, and R.L. Mercer. 1991. Word-Sense Disambiguation Using Statistical Methods. *ACL 1991*: 264-270

Gale, W.A., K.W. Church, and D. Yarowsky. 1992. A method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities* 26:415-439

Gale, W., K. Church, and D. Yarowsky. 1992. One Sense Per Discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*: 233-237.

Hirst, G. 1987. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge

Kelly, E. and P. Stone 1975. *Computer Recognition of English Word Senses*. North-Holland, Amsterdam.

Lin, D.K. 1997. Using Syntactic Dependency as Local Context to Resolve Word Sense Ambiguity. *ACL 1997*.

Lin, D.K. 1998. Automatic Retrieval and Clustering of Similar Words. *COLING-ACL 1998*.

Lin, D.K. 2002. Concept Discovery from Text. *COLING 2002*.

Press, W.H., S.A. Teukolsky, W. T. Vetterling, and B.P. Flannery. 1993. *Numerical Recipes in C*. Cambridge University Press

Rasmussen, E. 1992. Clustering Algorithm. In *Information Retrieval: Data Structure and Algorithms*, Frakes and Baeza-Yates (Eds.): 419-442, New Jersey: Prentice Hall.

Resnik, P. 1997. Selectional preference and sense disambiguation. *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?*

Schutze, H. Automatic Word Sense Disambiguation. *Computational Linguistics* 23 p97-124. 1998

Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Method. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.*