Using Semantic Similarity to Acquire Cooccurrence Restrictions from Corpora

Antonio Sanfilippo
SHARP Laboratories of Europe
Oxford Science Park
Oxford OX4 4GA, UK
antonio@sharp.co.uk

Abstract

We describe a method for acquiring semantic cooccurrence restrictions for tuples of syntactically related words (e.g. verb-object pairs) from text corpora automatically. This method uses the notion of semantic similarity to assign a sense from a dictionary database (e.g. WordNet) to ambiguous words occurring in a syntactic dependency. Semantic similarity is also used to merge disambiguated word tuples into classes of cooccurrence restrictions. This encoding makes it possible to reduce subsequent disambiguation events to simple table lookups.

1 Introduction

Although the assessment of semantic similarity using a dictionary database as knowledge source has been recognized as providing significant cues for word clustering (Resnik 1995b) and the determination of lexical cohesion (Morris & Hirst, 1991), its relevance for word disambiguation in running text remains relatively unexplored. The goal of this paper is to investigate ways in which semantic similarity can be used to address the disambiguation of syntactic collocates with specific reference to the automatic acquisition of semantic cooccurrence restrictions from text corpora.

A variety of methods have been proposed to rate words for semantic similarity with reference to an existing word sense bank. In Rada *et al.* (1989), semantic similarity is evaluated as the shortest path connecting the word senses being compared in a hierarchically structured thesaurus. Kozima & Furugori (1993) measure conceptual distance by spreading activation on a semantic network derived from LDOCE. Resnik (1995a) defines the semantic similarity between two words as the entropy value of the most informative concept subsuming the two words in a hierarchically structured thesaurus. A comparative assessment of these methods falls outside the scope of this paper as the approach to disambiguation we propose is in principle compatible with virtually any treatment of semantic similarity. Rather, our objective is to show that given a reliable calculation of semantic similarity, good results can be obtained in the disambiguation of words in context. In the work described here, Resnik's approach was used.

Following Resnik, semantic similarity is assessed with reference to the WordNet lexical database (Miller, 1990) where word senses are hierarchically structured. For example, (all senses of) the nouns *clerk* and *salesperson* in WordNet are connected to the first sense of the nouns *employee, worker, person* so as to indicate that *clerk* and *salesperson* are a kind of *employee* which is a kind of *worker* which in turn is a kind of *person*. In this case, the semantic similarity between the words *clerk* and *salesperson* would correspond to the entropy value of *employee* which is the most informative (i.e. most specific) concept shared by the two words. Illustrative extracts of WordNet with specific reference to the examples used throughout the paper are provided in table 1.

The information content (or entropy) of a concept c --- which in WordNet corresponds to a set of such as fire_v_4, dismiss_v_4, terminate_v_4, sack_v_2 --- is formally defined as $-log\ p(c)$ (Abramson, 1963:6-13). The probability of a concept c is obtained for each choice of text corpus or corpora collection K by dividing the frequency of c in K by the total number of words W observed in K which have the same part of speech p as the word senses in c:

$$(1)\ prob(c_p) = \frac{freq(c_p)}{W_p}$$

The frequency of a concept is calculated by counting the occurrences of all words which are potential instances of (i.e. subsumed by) the concept. These include words which have the same orthography and part of speech as the synonyms defining the concept as well as the concept's superordinates. Each time a word W_p is encountered in K, the count of each concepts c_p subsuming W_p (in any of its senses) is increased by one:

$$(2)\ freq(c_p) = \sum_{c_p \in \{x \mid sub(x,W_p)\}} count(W_p)$$

The semantic similarity between two words $W1_p$ $W2_p$ is expressed as the entropy value of the most informative concept c_p which subsumes both $W1_p$ and $W2_p$, as shown in (3).

$$(3)\ sim(W1_p, W2_p) = \max_{c_p \in \{x \mid sub(x, W1_p) \wedge sub(x, W2_p)\}} [-\log p(c_p)]$$

The specific senses of $W1_p$ $W2_p$ under which semantic similarity holds is determined with respect to the subsumption relation linking c_p with $W1_p$ $W2_p$. Suppose for example that in calculating the semantic similarity of the two verbs *fire, dismiss* using the WordNet lexical database we find that the most informative subsuming concept is represented by the synonym set containing the word sense remove_v_2. We will then know that the senses for *fire, dismiss* under which the similarity holds are fire_v_4 and dismiss_v_4 as these are the only instances of the verbs *fire* and *dismiss* subsumed by remove_v_2 in the WordNet hierarchy.

We propose to use semantic similarity to disambiguate syntactic collocates and to merge disambiguated collocates into classes of cooccurrence restrictions. Disambiguation of syntactic collocates results from intersecting pairs consisting of (*i*) a cluster containing all senses of a word collocate W1 having appropriate syntactic usage, and (*ii*) a cluster of semantically similar word senses related to W1 by the same syntactic dependency, e.g.:

(4) IN: < {fire_v_2/3/4/6/7/8},
{clerk_n_1/2, employee_n_1} >
< {fire_v_2/3/4/6/7/8},
{gun_n_1, rocket_n_1} >
< {hire_v_3, recruit_v_2},
{clerk_n_1/2} >
< {dismiss_v_4, fire_v_4},
{clerk_n_1/2} >

OUT: < {fire_v_4} , {clerk_n_1/2} >

The results of distinct disambiguation events are merged into pairs of semantically compatible word clusters using the notion of semantic similarity.

2 Extraction of Syntactic Word Collocates from Corpora

First, all instances of the syntactic dependency pairs under consideration (e.g. verb-object, verb-subject, adjective-noun) are extracted from a collection of text corpora using a parser. In performing this task, only the most important words (e.g. heads of immediate constituents) are chosen. The chosen words are also lemmatized. For example, the extraction of verb-object collocates from a text fragment such as *have certainly hired the best financial analysts in the area* would yield the pair < hire, analyst >.

The extracted pairs are sorted according to the syntactic dependency involved (e.g. verb-object). All pairs which involve the same dependency and share one word collocate are then merged. Each new pair consists of a unique **associating** word and a set of **associated** words containing all "statistically relevant" words (see below) which are related to the **associating** word by the same syntactic dependency, e.g.

(5) IN: < fire_v, gun_n >
< fire_v, rocket_n >
< fire_v, employee_n >
< fire_v, clerk_n >
< fire_v, hymn_n >
< fire_v, rate_n >
OUT: < fire_v,
{gun_n,rocket_n,employee_n,clerk_n} >

IN: < fire_v, employee_n >
< dismiss_v, employee_n >
< hire_v, employee_n >
< recruit_v, employee_n >
< attract_v, employee_n>
< be_v, employee_n>
< make_v, employee_n>
< affect_v, employee_n>
OUT: < {fire_v,dismiss_v,hire_v,recruit_v},
employee_n >

The statistical relevance of **associated** words is defined with reference to their conditional probability. For example, consider the equations in (6) where the numeric values express the (conditional) probability of occurrence in some corpus for each verb in (5) given the noun *employee*.

(6) *freq*(fire_v | employee_n) = .3
freq(dismiss_v | employee_n) = .28
freq(hire_v | employee_n) = .33
freq(recruit_v | employee_n) = .22
freq(attract_v | employee_n) = .02
freq(be_v | employee_n) = .002
freq(make_v | employee_n) = .005
freq(affect_v | employee_n>) = .01

These conditional probabilities are obtained by dividing the number of occurrences of the verb with *employee* by the total number of occurrences of the verb with reference to the text corpus under consideration, as indicated in (7).

$$(7)\ prob(W1 \mid W2) = \frac{count(W1, W2)}{count(W1)}$$

Inclusion in the set of statistically relevant associated words is established with reference to a threshold T1 which can be either selected manually or determined automatically as the most ubiquitous probability value for each choice of **associating** word. For example, the threshold T1 for the selection of verbs taking the noun *employee* as direct object with reference to the conditional probabilities in (6) can be calculated as follows. First, all probabilities in (6) are distributed over a ten-bin template, where each bin is to receive progressively larger values starting from a fixed lowest point greater than 0, e.g.:

From >	09	1	2	3	4	5	6	7	8	9
To	1	2	3	4	5	6	7	8	9	1
Values	--	--	3 28 22	33	--	--	--	--	--	--

Then one of the values from the bin containing most elements (e.g. the lowest) is chosen as the threshold. The exclusion of collocates which are not statistically relevant in the sense specified above makes it possible to avoid interference from collocations which do not provide sufficiently specific exemplifications of word usage.

3 Word Clustering and Sense Expansion

Each pair of syntactic collocates at this stage consists of either

- an **associating** head word (AING) and a set of dependent **associated** words (AED), e.g.

  < AING: fire_v,
  AED: {gun_n,rocket_n,employee_n,clerk_n} >

- or an **associating** dependent word (AING) and a set of **associated** head words (AED), e.g.

  < AED: {fire_v,dismiss_v,hire_v,recruit_v},
  AING: employee_n>

The next step consists in partitioning the set of associated words into clusters of semantically congruent word senses. This is done in three stages.

1. Form all possible unique word pairs with non-identical members out of each associated word set, e.g.

   IN: {fire, dismiss, hire, recruit}
   OUT: {fire-dismiss,fire-hire,fire-recruit,
   dismiss-hire,dismiss-recruit,
   hire-recruit}

   IN: {gun,rocket,employee,clerk}
   OUT: {gun-rocket,gun-employee,
   gun-clerk,rocket-employee,
   rocket-clerk,employee-clerk}

2. Find the semantic similarity (expressed as a numeric value) for each such pair, specifying the senses with respect to which the similarity holds (if any), e.g.

   IN: {fire-dismiss,fire-hire,fire-recruit,
   dismiss-hire,dismiss-recruit,hire-recruit}
   OUT: {*sim*(fire_v_4,dismiss_v_4) = 6.124,
   sim(fire,hire) = 0,
   sim(fire,recruit) = 0,
   sim(dismiss,hire) = 0,
   sim(dismiss,recruit) = 0,
   sim(hire_v_3,recruit_v_2) = 3.307}

   IN: {gun-rocket,gun-employee,
   gun-clerk,rocket-employee,
   rocket-clerk,employee-clerk}
   OUT: {*sim*(gun_n_1,rocket_n_1) = 5.008,
   sim(gun_n_1-3,employee_n_1) = 1.415,
   sim(gun_n_1-3,clerk_n_1/2) = 1.415,
   sim(rocket_n_3,employee_n_1) = 2.255,
   sim(rocket_n_3,clerk_n_1/2) = 2.255,
   sim(employee_n_1,clerk_n_1/2) = 4.144}

   The assessment of semantic similarity and the ensuing word sense specification are carried out using Resnik's approach (see section **1**).

3. Fix the threshold for membership into clusters of semantically congruent word senses (either manually or by calculation of the most ubiquitous semantic similarity value) and generate such clusters. For example, assuming a threshold value of 3, we will have:

   IN: {*sim*(fire_v_4,dismiss_v_4) = 6.124,
   sim(fire,hire) = 0,
   sim(fire,recruit,) = 0,
   sim(dismiss,hire) = 0,
   sim(dismiss,recruit) = 0,
   sim(hire_v_3,recruit_v_2) = 3.307}
   OUT: {fire_v_4,dismiss_v_4}
   {hire_v_3,recruit_v_2}

   IN: {*sim*(gun_n_1,rocket_n_1) = 5.008,
   sim(gun_n_1/2/3,employee_n_1) = 1.415,
   sim(gun_n_1/2/3,clerk_n_1/2) = 1.415,
   sim(rocket_n_3,employee_n_1) = 2.255,
   sim(rocket_n_3,clerk_n_1/2) = 2.255,
   sim(employee_n_1,clerk_n_1/2) = 4.144}
   OUT: {clerk_n_1/2,employee_n_1}
   {gun_n_1,rocket_n_1}

   Once associated words have been partitioned into semantically congruent clusters, new sets of collocations are generated as shown in (8) by

   - pairing each cluster of semantically congruent associated words with its associating word, and
   - expanding the associating word into all of its possible senses.

   At this stage, all word senses which are syntactically incompatible with the original input words are removed. For example, the intransitive verb senses fire_v_1 and fire_v_5 (see table 1) are eliminated since the occurrence of *fire* in the input collocation which we are seeking to disambiguate relates to the transitive use of the verb. Note that the noun employee has only one sense in WordNet (see table 1); therefore, employee has a single expansion when used as an **associating** word.

(8) IN: < AED: { {hire_v_3,recruit_v_2},
{dismiss_v_4,fire_v_4} },
AING: employee_n >
OUT: < {hire_v_3,recruit_v_2}, {employee_n_1} >
< {dismiss_v_4,fire_v_4}, {employee_n_1} >

IN: < AING:fire_v,
AED:{ {clerk_n_1,clerk_n_2,employee_n_1},
{gun_n_1,rocket_n_1} } >
OUT: < {fire_v_2/3/4/6/7/8},
{clerk_n_1/2,employee_n_1} >
< {fire_v_2/3/4/6/7/8},
{gun_n_1,rocket_n_1} >

4 Disambiguating the "Associating" Word and Merging Disambiguated Collocations

The disambiguation of the associating word is performed by intersecting correspondent subsets across pairs of the newly generated collocations. In the case of verb-object pairs, for example, the subsets of these new sets containing verbs are intersected and likewise the subsets containing objects are intersected. The output comprises a new set which is non-empty if the two sets have one or more common members in both the verb and object subsets. For the specific example of newly expanded collocations given in (8), there is only one pairwise intersection producing a non empty result, as shown in (9).

(9) IN: < {fire_v_2/3/4/6/7/8},
{clerk_n_1/2,employee_n_1} >
< {dismiss_v_4,fire_v_4},
{employee_n_1} >

OUT: < {fire_v_4}, {employee_n_1} >

All other pairwise intersections are empty as there are no verbs and objects common to both sets of each pairwise combination.

The result of distinct disambiguation events can be merged into pairs of semantically compatible word clusters using the notion of semantic similarity. For example, the verbs and nouns of all the input pairs in (10) are closely related in meaning and can therefore be merged into a single pair.

(10) IN: < fire_v_4 , employee_n_1 >
< dismiss_v_4, clerk_n_1 >
< give_the_axe_v_1 , salesclerk_n_1 >
< sack_v_2 , shop_clerk_n_1 >
< terminate_v_4, clerk_n_2 >

OUT: < {fire_v_4, dismiss_v_4, sack_v_2,
give_the_axe_v_1, terminate_v_4},
{clerk_n_1, employee_n_1, clerk_n_2
salesclerk_n_1, shop_clerk_n_1} >

5 Storing Results

Pairs of semantically congruent word sense clusters such as the one shown in the output of (10) are stored as cooccurrence restrictions so that future disambiguation events involving any head-dependent word sense pair in them can be reduced to simple table lookups.

The storage procedure is structured in three phases. First, each cluster of word senses in each pair is assigned a unique code consisting of an id number and the syntactic dependency involved:

(11) < {102_VO, fire_v_4, dismiss_v_4, sack_v_2,
give_the_axe_v_1, send_away_v_2,
force_out_v_2, terminate_v_4},
{102_OV, clerk_n_1/2, employee_n_1,
salesclerk_n_1, shop_clerk_n_1} >

< {103_VO, lease_v_4, rent_v_3, hire_v_3,
charter_v_3, engage_v_6, take_v_22,
recruit_v_2},
{102_OV, clerk_n_1/2, employee_n_1,
salesclerk_n_1, shop_clerk_n_1} >

< {104_VO, shoot_v_3, fire_v_1, ...},
{104_OV, gun_n_1, rocket_n_1, ...} >

Then, the cluster codes in each pair are stored in a cooccurrence restriction table:

102_VO	102_OV
103_VO	103_OV
104_VO	104_OV

Finally, each word sense is stored along with its associated cluster code(s):

fire_v_4	102_VO
dismiss_v_4	102_VO
clerk_n_1/2	102_VO
employee_n_1	102_VO
hire_v_3	103_VO
recruit_v_2	102_VO
shoot_v_3	104_VO
fire_v_1	104_VO
gun_n_1	104_VO
rocket_n_1	104_VO

The disambiguation of a pair of syntactically related words such as the pair <fire_v, employee_n> can be carried out by

- retrieving all the cluster codes for each word in the pair and create all possible pairwise combinations, e.g.

IN: < fire_v , employee_n >
OUT: < 102_VO, 102_OV >
< 104_VO, 102_OV >

- eliminating code pairs which are not in the table of cooccurrence restrictions for cluster codes, e.g.

  INPUT: < 102_VO, 102_OV >
  < 104_VO, 102_OV >
  OUTPUT: < 102_VO, 102_OV >

- using the resolved cluster code pairs to retrieve the appropriate senses of the input words from previously stored pairs of word senses and cluster codes such as those in the table above, e.g.

  INPUT: < [fire_v, 102_VO] ,
  [employee_n, 102_OV] >
  OUTPUT: < fire_v_4, employee_n_1 >

By repeating the acquisition process described in sections **2-4** for collections of appropriately selected source corpora, the acquired cooccurrence restrictions can be parameterized for sublanguage specific domains. This augmentation can be made by storing each word sense and associated cluster code with a sublanguage specification and a percentage descriptor indicating the relative frequency of the word sense with reference to the cluster code in the specified sublanguage, e.g.

fire_v_4	102_VO	Business	65%
fire_v_4	102_VO	Crime	25%
fire_v_1	104_VO	Business	5%
fire_v_1	104_VO	Crime	70%

6 Statistically Inconspicuous Collocates

Because only statistically relevant collocations are chosen to drive the disambiguation process (see section **2**), it follows that no cooccurrence restrictions will be acquired for a variety of word pairs. This, for example, might be the case with verb-object pairs such as < fire_v, hand_n > where the noun is a somewhat atypical object. This problem can be addressed by using the cooccurrence restrictions already acquired to classify statistically inconspicuous collocates, as shown below with reference to the verb object pair < fire_v, hand_n >.

- Find all verb-object cooccurrence restrictions containing the verb *fire*, which as shown in the previous section are

  < 102_VO, 102_OV >
  < 104_VO, 104_OV >

- Retrieve all members of the direct object collocate class, e.g.

  102_OV -> clerk_n_1/2, employee_n_1
  104_OV -> gun_n_1, rocket_n_1

Cluster the statistically inconspicuous collocate with all members of the direct object collocate class. This will provide one or more sense classifications for the statistically inconspicuous collocate. In the present case, the WordNet senses 2 and 9 (glossed as "farm labourer" and "crew member" respectively) are given when hand_n clusters with clerk_n_1/2 and employee_n_1, e.g.

IN: {hand_n, clerk_n_1/2, employee_n_1,
gun_n_1, rocket_n_1}
OUT: {hand_n_2/9, clerk_n_1/2, employee_n_1}
{gun_n_1, rocket_n_1}

- Associate the disambiguated statistically inconspicuous collocate with the same code of the word senses with which it has been clustered, e.g.

hand	n	2	102_VO
hand	n	9	102_VO

This will make it possible to choose senses 2 and 9 for *hand* in contexts where *hand* occurs as the direct object of verbs such as *fire*, as explained in the previous section.

7 Preliminary Results and Future Work

A prototype of the system described was partially implemented to test the effectiveness of the disambiguation method. The prototype comprises:

- a component performing semantic similarity judgements for word pairs using WordNet (this is an implementation of Resnik's approach);

- a component which turns sets of word pairs rated for semantic similarity into clusters of semantically congruent word senses, and

- a component which performs the disambiguation of syntactic collocates in the manner described in section **4**.

The current functionality provides the means to disambiguate a pair of words <W1 W2> standing in a given syntactic relation *Dep* given a list of words related to W1 by *Dep*, a list of words related to W2 by *Dep*, and a semantic similarity threshold for word clustering, as shown in (12).

In order to provide an indication of how well the system performs, a few examples are presented in (12). As can be confirmed with reference to the WordNet entries in table 1, these preliminary results are encouraging as they show a reasonable resolution of ambiguities. A more thorough evaluation is currently being carried out.

(12) IN: < fire_v-[employee_n,clerk_n, gun_n,pistol_n],
[fire,dismiss, hire,recruit]-employee_n, 3 >
OUT: < fire_v_4 employee_n_1 >

IN: < fire_v-[employee_n,clerk_n, gun_n,pistol_n],
[fire_v,shoot_v, pop_v,disharge_v]-gun_n, 3 >
OUT: < fire_v_1 gun_n_1 >

IN: < wear_v-[suit_n,garment_n, clothes_n,uniform_n],
[wear_v, have_on_v, record_v,file_v]-suit_n, 3 >
OUT. < wear_v_1/9 suit_n_1 >

IN. < file_v-[suit_n,proceedings_n, lawsuit_n,
litigation_n],
[wear,have_on, record_v,file_v]-suit_n, 3 >
OUT. < file_v_1/5 suit_n_2 >

Note that disambiguation can yield multiple senses, as shown with reference to the resolution of the verbs *file* and *wear* in the third and fourth examples shown in (12). Multiple disambiguation results typically occur when some of the senses given for a word in the source dictionary database are close in meaning. For example, both sense 1 and 9 of *wear* relate to an eventuality of "clothing oneself". Multiple word sense resolutions can be ranked with reference to the semantic similarity scores used in clustering word senses during disambiguation. The basic idea is that the word sense resolution contained in the word cluster which has highest semantic similarity scores provides the best disambiguation hypothesis. For example, specific word senses for the verb-object pair < wear suit > in the third example of (12) above are given by the disambiguated word tuples in (13) which arise from intersecting pairs consisting of all senses of an **associating** word and a semantically congruent cluster of its **associated** words, as described in section 4.

(13) { < {have_on_v_1,wear_v_1},
{clothes_n_1,garment_n_1, suit_n_1,
uniform_n_1} >
< {file_v_2,wear_v_9},
{clothes_n_1,garment_n_1, suit_n_1,
uniform_n_1} > }

Taking into account the scores shown in (14), the best word sense candidate for the verb *wear* in the context *wear suit* would be wear_v_1. In this case, the semantic similarity scores for the second cluster (i.e. the nouns) do not matter as there is only one such cluster.

(14) *sim*(have_on_v_1, wear_v_1) = 6.291
sim(file_v_2, wear_v_9) = 3.309

Preliminary results suggest that the present treatment of disambiguation can achieve good results with small quantities of input data. For example, as few as four input collocations may suffice to provide acceptable results, e.g.

(15) IN: < fire_v-[employee_n,clerk_n],
[fire,dismiss]-employee_n, 3 >
OUT: < fire_v_4 employee_n_1 >

IN: < wear_v-[suit_n,clothes_n],
[wear_v,have_on_v]-suit_n, 3 >
OUT: < wear_v_1 suit_n_1 >

This is because word clustering --- which is the decisive step in disambiguation --- is carried out using a measure of semantic similarity which is essentially induced from the hyponymic links of a semantic word net. As long as the collocations chosen as input data generate some word clusters, there is a good chance for disambiguation. The reduction of input data requirements offers a significant advantage compared with methods such as those presented in Brown *et al.* (1991), Gale *et al.* (1992), Yarowsky (1995), and Karol & Edelman (1996) where strong reliance on statistical techniques for the calculation of word and context similarity commands large source corpora. Such advantage can be particularly appreciated with reference to the acquisition of cooccurrence restrictions for those sublanguage domains where large corpora are not available.

Ironically, the major advantage of the approach proposed --- namely, a reliance on structured semantic word nets as the main knowledge source for assessing semantic similarity --- is also its major drawback. Semantically structured lexical databases, especially those which are tuned to specific sublanguage domains, are currently not easily available and expensive to build manually. However, advances in the area of automatic thesaurus discovery (Grefenstette, 1994) as well as progress in the area of automatic merger of machine readable dictionaries (Sanfilippo & Poznanski, 1992; Chang & Chen, 1997) indicate that availability of the lexical resources needed may gradually improve in the future. In addition, ongoing research on rating conceptual distance from unstructured synonym sets (Sanfilippo, 1997) may soon provide an effective way of adapting any commercially available thesaurus to the task of word clustering, thus increasing considerably the range of lexical databases used as knowledge sources in the assessment of semantic similarity.

Acknowledgements

This research was carried out within the SPARKLE project (LE-12111). I am indebted to Geert Adriaens, Simon Berry, Ted Briscoe, Ian Johnson, Victor Poznanski, Karen Sparck Jones, Ralf Steinberger and Yorick Wilks for valuable feedback.

References

N. Abramson. 1963. *Information Theory and Coding.* McGraw-Hill, NY.

P. Brown, S. Pietra, V. Pietra & R. Mercer. 1991. Word sense disambiguation using statistical methods. In *Proceedings of ACL*, pp. 264-270.

J. Chang and J. Chen. 1997. Topical Clustering of MRD Senses based on Information Retrieval Techniques. Ms. Dept. of Computer Science, National Tsing Hua University, Taiwan.

W. Gale, K. Church & D. Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415-439.

G. Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston.

Y. Karov & S. Edelman. 1993. Learning similarity-based word sense disambiguation from sparse data. Available at **`Fout! Bladwijzer niet gedefinieerd.`** as paper No. 9605009

H. Kozima & T. Furugori. 1993. Similarity between Words Computed by Spreading Activation on an English Dictionary. *In Proceedings of EACL.*

G. Miller. 1990 Five Papers on WordNet. Special issue of the *International Journal of Lexicography*, 3(4).

J. Morris & G. Hirst. 1991. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics*, 17:21-48.

R. Rada, M. Hafedh, E. Bicknell and M. Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on System, Man, and Cybernetics*, 19(1):17-30.

P. Resnik. 1995a. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI.*

P. Resnik. 1995b. Disambiguating noun groupings with respect to WordNet Senses. In *Proceedings of 3rd Workshop on Very Large Corpora.* Association for Computational Linguistics.

A. Sanfilippo. 1997. Rating conceptual distance using extended synonym sets. Ms. SHARP Lab. of Europe, Oxford.

A. Sanfilippo and V. Poznanski. 1992. The Acquisition of Lexical Knowledge from Combined Machine-Readable Dictionary Sources. *In Proceedings of the 3rd Conference on Applied Natural Language Processing,* Trento.

D. Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *In Proceedings of the 33rd Annual Meeting of the ACL*, pp. 189-96.

dismiss	v	1	*She dismissed his advances*
		2	put out of judicial consideration
		3	stop associating with
		4	give notice
		5	end one's encounter with somebody by causing or permitting the person to leave
file	v	1	register in a public office or in a court of law
		2	**smooth with a file**
		3	proceed in file
		4	file a formal charge against
		5	place in a file
fire	v	1	**open fire**
		2	fire a gun, fire a bullet
		3	of pottery
		4	give notice
		5	*The gun fired*
		6	Call forth, of emotions, feelings, and responses
		7	
		8	*They burned the house and his diaries*
			provide with fuel
hire	v	1	
		2	engage or hire for work
		3	of goods and services
			engage in a commercial transaction
recruit	v	1	
		2	register formally, as a participant or member
		3	*The lab director recruited an able crew of assistants*
			conscript, levy
wear	v	1	
		2	be dressed in
		3	*He wore a red ribbon*
		4	have in one's aspect
		5	*Wear one's hair in a certain way*
		6	**hold out, endure**
		7	**wear off, wear out, wear thin**
		8	go to pieces
		9	wear out
			put clothing on one's body
employee	n	1	
			a worker who is hired to perform a job
clerk	n	1	keeps records or accounts
		2	a salesperson in a store
			a salesperson in a store
gun	n	1	
		2	a weapon that discharges a missile
		3	large but transportable guns
			a pedal or hand-operated lever that controls the throttle
rocket	n	1	
		2	any weapon propelled by a rocket engine
		3	a device containing its own propellant and driven by reaction propulsion
		4	erect European annual often grown as a salad crop to be harvested when young and tender
		5	propels bright light high in the sky, or used to propel a lifesaving line or harpoon
			sends a firework display high into the sky
suit	n	1	
			a set of garments for outerwear all of the same
		2	fabric and color

			a judicial proceeding brought by one party
		3	against another
		4	**suing**
			any of four sets of 13" cards in a pack

Table 1 Extract entries of the WordNet Lexical Database. Synonyms (in boldface) and examples (in italic) are omitted unless used in place of definitions. Other thesaural relations -- -e.g. hyponymy, holonymy, etc. --- are also omitted.