

Inherited Feature-based Similarity Measure Based on Large Semantic Hierarchy and Large Text Corpus

Hideki Hirakawa

Toshiba R&D Center

1 Komukai Toshiba-cho, Saiwai-ku,

Kawasaki 210, JAPAN

hirakawa@eel.rdc.toshiba.co.jp

Zhonghui Xu, Kenneth Haase

MIT Media Laboratory

20 Ames Street

Cambridge, MA 02139 USA

{xu,haase}@media.mit.edu

Abstract

We describe a similarity calculation model called IFSM (Inherited Feature Similarity Measure) between objects (words/concepts) based on their common and distinctive features. We propose an implementation method for obtaining features based on abstracted triples extracted from a large text corpus utilizing taxonomical knowledge. This model represents an integration of traditional methods, i.e., relation based similarity measure and distribution based similarity measure. An experiment, using our new concept abstraction method which we call the flat probability grouping method, over 80,000 surface triples, shows that the abstraction level of 3000 is a good basis for feature description.

1 Introduction

Determination of semantic similarity between words is an important component of linguistic tasks ranging from text retrieval and filtering, word sense disambiguation or text matching. In the past five years, this work has evolved in conjunction with the availability of powerful computers and large linguistic resources such as WordNet (Miller,90), the EDR concept dictionary (EDR,93), and large text corpora.

Similarity methods can be broadly divided into "relation based" methods which use relations in an ontology to determine similarity and "distribution based" methods which use statistical analysis as the basis of similarity judgements. This article describes a new method of similarity matching, *inherited feature based similarity matching* (IFSM) which integrates these two approaches.

Relation based methods include both depth based and path based measures of similarity. The *Most Specific Common Abstraction* (MSCA) method compares two concepts based on the taxonomic depth of their common parent; for example, "dolphin" and "human" are more similar than "oak" and "human" because the common concept "mammal" is deeper in the taxonomy than "living thing".

Path-length similarity methods are based on counting the links between nodes in a semantic network. (Rada,89) is a widely adopted approach to such matching and (Sussna,93) combines it with WordNet to do semantic disambiguation.

The chief problems with relation-based similarity methods lie in their sensitivity to artifacts in the coding of the ontology. For instance, MSCA algorithms are sensitive to the relative depth and detail of different parts of the concept taxonomy. If one conceptual domain (say plants) is sketchily represented while another conceptual domain (say, animals) is richly represented, similarity comparisons within the two domains will be incommensurable. A similar problem plagues path-length based algorithms, causing nodes in richly structured parts of the ontology to be consistently judged less similar to one another than nodes in shallower or less complete parts of the ontology.

Distribution-based methods are based on the idea that the similarity of words can be derived from the similarity of the contexts in which they occur. These methods differ most significantly in the way they characterize contexts and the similarity of contexts. Word Space (Schutze,93) uses letter 4-grams to characterize both words and the contexts in which they appear. Similarity is based on 4-grams in common between the contexts. Church and Hanks ('89) uses a word window of set size to characterize the context of a word based on the immediately adjacent words. Other methods include the use of expensive-to-derive features such as subject-verb-object (SVO) relations (Hindle,90) or other grammatical relations (Grefenstette,94). These choices are not simply implementational but imply different similarity judgements. The chief problem with distribution based methods is that they only permit the formation of first-order concepts definable directly in terms of the original text. Distribution based methods can acquire concepts based on recurring patterns of words but not on recurring patterns of concepts. For instance, a distributional system could easily identify that an article involves lawyers based on recurring instances of words like "sue" or "court". But it could not use the occurrence of these concepts as conceptual cues for

developing concepts like "litigation" or "pleading" in connection with the "lawyer" concept.

One notable integration of relation based and distributional methods is Resnik's annotation of a relational ontology with distributional information (Resnik,95a,95b). Resnik introduces a "class probability" associated with nodes (synsets) in WordNet and uses these to determine similarity. Given these probabilities, he computes the similarity of concepts based on the "information" that would be necessary to distinguish them, measured using information-theoretic calculations.

2 The Feature-based Similarity Measure

The Inherited Feature Similarity Measure (IFSM) is another integrated approach to measuring similarity. It uses a semantic knowledge base where concepts are annotated with *distinguishing features* and bases similarity on comparing these sets of features. In our experiments, we derived the feature sets by a distributional analysis of a large corpus.

Most existing relation-based similarity methods directly use the relation topology of the semantic network to derive similarity, either by strategies like link counting (Rada,89) or the determination of the depth of common abstractions (Kolodner,89). IFSM, in contrast, uses the topology to derive descriptions whose comparison yields a similarity measure. In particular, it assumes an ontology where:

1. Each concept has a set of features
2. Each concept inherits features from its generalizations (hypernyms)
3. Each concept has one or more "distinctive features" which are not inherited from its hypernyms.

Note that we neither claim nor require that the features completely characterize their concepts or that inheritance of features is sound. We only require that there be some set of features we use for similarity judgements. For instance, a similarity judgement between a penguin and a robin will be partially based on the feature "can-fly" assigned to the concept bird, even though it does not apply individually to penguins.

Fig 1 shows a simple example of a fragment of a conceptual taxonomy with associated features. Inherited features are in *italic* while distinctive

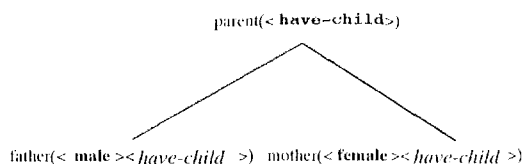


Fig.1 Fragment of conceptual taxonomy

features are in bold. In our model, features have a weight based on the importance of the feature to the concept.

We have chosen to automatically generate features distributionally by analyzing a large corpus. We describe this generation process below, but we will first turn to the evaluation of similarity based on featural analysis.

2.1 Approaches to Feature Matching

There are a variety of similarity measures available for sets of features, but all make their comparisons based on some combination of shared features, distinct features, and shared absent features (e.g., neither X or Y is red). For example, Tversky ('77) proposes a model (based on human similarity judgements) where similarity is a linear combination of shared and distinct features where each feature is weighted based on its importance. Tversky's experiment showed the highest correlation with human subjects' feelings when weighted shared and distinct features are taken into consideration.

SEXTANT (Grefenstette,94) introduced the Weighted Jaccard Measure which combines the Jaccard Measure with weights derived from an information theoretic analysis of feature occurrences. The weight of a feature is computed from a global weight (based on the number of global occurrences of the word or concept) and a local weight (based on the frequency of the features attached to the word).

In our current work, we have adopted the Weighted Jaccard Measure for preliminary evaluation of our approach. The distinctive feature of our approach is the use of the ontology to derive features rather than assuming atomic sets of features.

2.2 Properties of IFSM

In this section we compare IFSM's similarity judgements to those generated by other methods. In our discussion, we will consider the simple network of Fig 2. We will use the expression $sim(c_i, c_j)$ to denote the similarity of concepts c_1 and c_2 .

Given the situation of Fig 2, both MSCA and Resnik's MISM (Most Informative Subsumer Method) assert $sim(C1, C2) = sim(C2, C3)$. MSCA makes the similarity the same because they have the same (nearest) common abstraction C0. MISM holds the similarity to be the same because

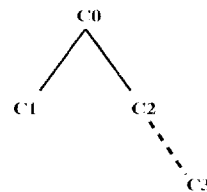
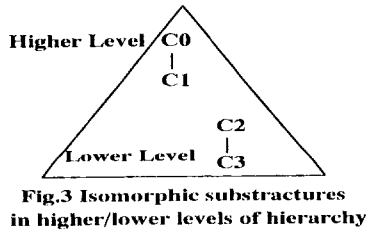


Fig.2 Example of a hierarchical structure



the assertion of C2 adds no information given the assertion of C3. Path-length methods, in contrast, assert $sim(C1, C2) < sim(C2, C3)$ since the number of links between the concepts is quite different.

Because IFSM depends on the features derived from the network rather than on the network itself, judgements of similarity depend on the exact features assigned to C1, C2, and C3. Because IFSM assumes that some distinctive features exist for C3, $sim(C1, C2)$ and $sim(C1, C3)$ are unlikely to be identical. In fact, unless the distinctive features of C3 significantly overlap the distinctive feature of C1, it will be the case that $sim(C1, C2) < sim(C2, C3)$.

IFSM differs from the path length model because it is sensitive to depth. If we assume a relatively uniform distribution of features, the total number of features increases with depth in the hierarchy. This means that $sim(C0, C1)$ located in higher part of the hierarchy is expected to be less than $sim(C2, C3)$ located in lower part of the hierarchy.

3 Components of IFSM model

IFSM consists of a hierarchical conceptual thesaurus, a set of distinctive features assigned to each object and weightings of the features. We can use, for example, WordNet or the EDR concept dictionary as a hierarchical conceptual thesaurus. Currently, there are no explicit methods to determine sets of distinctive features and their weightings of each object (word or concept).

Here we adopt an automatic extraction of features and their weightings from a large text corpus. This is the same approach as that of the distributed semantic models. However, in contrast to those models, here we hope to make the level of the representation of features high enough to capture semantic behaviors of objects.

For example, if one relation and one object can be said to describe the features of object, we can define one feature of "human" as "agent of walking". If more context is allowed, we can define a feature of "human" as "agent of utilizing fire". A wider context gives a precision to the contents of the features. However, a wider context exponentially increases the possible number of features which will exceed current limitations of computational resources. In consideration of these factors, we adopt triple relations such as "dog chase cat", "cut paper with scissors" obtained from the cor-

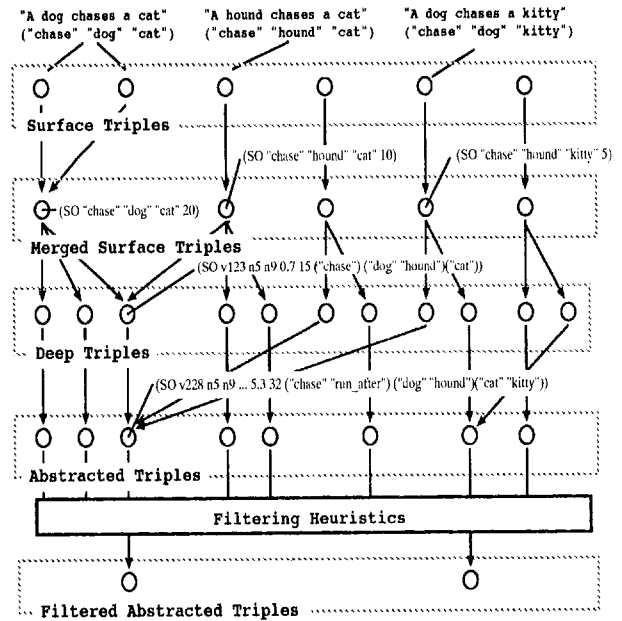


Fig.4 Abstracted triple extraction from corpus

pus as a resource of features, and apply class based abstraction (Resnik 95a) to triples to reduce the size of the possible feature space.

As mentioned above, features extracted from the corpus will be represented using synsets/concepts in IFSM. Since no large scale corpus data with semantic tags is available, the current implementation of IFSM has a word sense disambiguation problem in obtaining class probabilities. Our current basic strategy to this problem is similar to (Resnik,95a) in the sense that synsets associated with one word are assigned uniform frequency or "credit" when that word appears in the corpus. We call this strategy the "brute-force" approach, like Resnik. On top of this strategy, we introduce filtering heuristics which sort out unreliable data using heuristics based on the statistical properties of the data.

4 The feature extraction process

This section describes the feature extraction procedure. If a sentence "a dog chased a cat" appears in the corpus, features representing "chase cat" and "dog chase" may be attached to "dog" and "cat" respectively. Fig 4 shows the overall process used to obtain a set of abstracted triples which are sources of feature and weighting sets for synsets.

4.1 Extraction of surface typed triples from the corpus

Typed surface triples are triples of surface words holding some fixed linguistic relations (Hereafter call this simply "surface triples"). The current implementation has one type "SO" which represents

"subject - verb - object" relation. A set of typed surface triples are extracted from a corpus with their frequencies.

Surface triple set

(TYPE VERB NOUN1 NOUN2 FREQUENCY)

Ex. (SO "chase" "dog" "cat" 10)

4.2 Expansion of surface triples to deep triples

Surface triples are expanded to corresponding deep triples (triples of synset IDs) by expanding each surface word to its corresponding synsets. The frequency of the surface triples is divided by the number of generated deep triples and it is assigned to each deep triple. The frequency is also preserved as it is as an occurrence count. Surface words are also reserved for later processings.

Deep triple collection

(TYPE V-SYNSET N1-SYNSET N2-SYNSET FREQUENCY OCCURRENCE V-WORD N1-WORD N2-WORD)

Ex. (SO v123 n5 n9 0.2 10 "chase" "dog" "cat")

"v123" and "n5" are synset IDs corresponding to word "chase" and "dog" respectively. These deep triples are sorted and merged. The frequencies and the occurrence counts are summed up respectively. The surface words are merged into surface word lists as the following example shows.

Deep triple set

(TYPE V-SYNSET N1-SYNSET N2-SYNSET FREQUENCY OCCURRENCE V-WORD N1-WORD N2-WORDS)

Ex. (SO v123 n5 n9 0.7 15 ("chase")("dog" "hound")("cat"))

In this example, "dog" and "hound" have same synset ID "n9".

4.3 Synset abstraction method

The purpose of the following phases is to extract feature sets for each synset in an abstracted form. In an abstracted form, the size of each feature space becomes tractable.

Abstraction of a synset can be done by dividing whole synsets into the appropriate number of synset groups and determining a representative of each group to which each member is abstracted. There are several methods to decide a set of synset groups using a hierarchical structure. One of the simplest methods is to make groups by cutting the hierarchy structure at some depth from the root. We call this the flat-depth grouping method. Another method tries to make the number of synsets in a group constant, i.e., the upper/lower bound for a number of concepts is given as a criteria (Hearst,93). We call this the flat-size grouping method. In our implementation, we introduce a new grouping method called the flat-probability grouping method in which synset groups are specified such that every group has the same class probabilities. One of the advantages of this method is that it is expected to give a grouping based on the quantity of information which will be suitable for the target task, i.e., semantic abstraction of

triples. The degree of abstraction, i.e., the number of groups, is one of the principal factors in deciding the size of the feature space and the preciseness of the features (power of description).

4.4 Deep triple abstraction

Each synset of deep triples is abstracted based on the flat-probability grouping method. These abstracted triples are sorted and merged. Original synset IDs are maintained in this processing for feature extraction process. The result is called the abstracted deep triple set.

Abstracted deep triple set

(TYPE V-ABS-SYNSET N1-ABS-SYNSET N2-ABS-SYNSET V-SYNSET-LIST N1-SYNSET-LIST N2-SYNSET-LIST SYN-FREQUENCY OCCURRENCE V-WORDS N1-WORDS N2-WORDS)

Ex. (SO v28 n5 n9 (v123 v224) (n5) (n9 n8) 5.3 32 ("chase" "run" "after")("dog" "hound") ("cat" "kitty"))

Synset "v28" is an abstraction of synset "v123" and synset "v224" which corresponds to "chase" and "run_after" respectively. Synset "n9" corresponding to "cat" is an abstraction of synset "n8" corresponding to "kitty".

4.5 Filtering abstracted triples by heuristics

Since the current implementation adopts the "brute-force" approach, almost all massively generated deep triples are fake triples. The filtering process reduces the number of abstracted triples using heuristics based on statistical data attached to the abstracted triples. There are three types of statistical data available; i.e., estimated frequency, estimated occurrences of abstracted triples and lists of surface words.

Here, the length of a surface word list associated with an abstracted synset is called a surface support of the abstracted synset. A heuristics rule using some fixed frequency threshold and a surface support bound are adopted in the current implementation.

4.6 Common feature extraction from abstracted triple set

This section describes a method for obtaining features of each synset. Basically a feature is typed binary relation extracted from an abstracted triple. From the example triple,

(SO v28 n5 n9 (v123 v224) (n5) (n9 n8) 5.3 32 ("chase" "run" "after")("dog" "hound") ("cat" "kitty"))

the following features are extracted for three of the synsets contained in the above data.

n5 (ov v28 n9 5.3 32 ("chase" "run" "after")("cat" "kitty"))
n9 (sv v28 n5 5.3 32 ("chase" "run" "after")("dog" "hound"))
n8 (sv v28 n5 5.3 32 ("chase" "run" "after")("dog" "hound"))

An abstracted triple represents a set of examples in the text corpus and each sentence in the corpus usually describes some specific event. This means that the content of each abstracted

triple cannot be treated as generally or universally true. For example, even if a sentence "a man bit a dog" exists in the corpus, we cannot declare that "biting dogs" is a general property of "man". Metaphorical expressions are typical examples. Of course, the distributional semantics approach assumes that such kind of errors or noise are hidden by the accumulation of a large number of examples.

However, we think it might be a more serious problem because many uses of nouns seem to have an anaphoric aspect, i.e., the synset which best fits the real world object is not included in the set of synsets of the noun which is used to refer to the real world object. "The man" can be used to express any descendant of the concept "man". We call this problem the word-referent disambiguation problem. Our approach to this problem will be described elsewhere.

5 Preliminary experiments on feature extraction using 1010 corpus

In this section, our preliminary experiments of the feature extraction process are described. In these experiments, we examine the proper granularity of abstracted concepts. We also discuss a criteria for evaluating filtering heuristics. WordNet 1.4, 1010 corpus and Brown corpus are utilized through the experiments. The 1010 corpus is a multi-layered structured corpus constructed on top of the FRAMER-D knowledge representation language. More than 10 million words of news articles have been parsed using a multi-scale parser and stored in the corpus with mutual references to news article sources, parsed sentence structures, words and WordNet synsets.

5.1 Experiment on flat-probability grouping

To examine the appropriate number of abstracted synsets, we calculated three levels of abstracted synset sets using the flat probability grouping method. Class probabilities for noun and verb synsets are calculated using the brute force method based on 280K nouns and 167K verbs extracted from the Brown corpus (1 million words).

We selected 500, 1500, 3000 synset groups for candidates of feature description level. The 500 node level is considered to be a lowest boundary and the 3000 node level is expected to be the tar-

Depth	1	2	3	4	5	6	7	8
Synsets	611	122	966	2949	5745	12293	8384	7408
Depth	9	10	11	12	13	14	15	16
Synsets	5191	3068	1417	812	314	94	36	6

Table 1. Depth/Noun_Synsets in WordNet 1.4

Level 500 (518 synsets)

- 1 {structure construction}(7219.47 4): a thing constructed; a complex construction or entity
- 2 {time_period period period_of_time}(6934 3): a length of time; "government services began during the colonial period"
- 3 {organization}(6469.94 4): a group of people who work together
- 4 {action}(6370.54 2): something done;
- 5 {natural_object}(6277.26 3): an object occurring naturally;

Level 3000 (3001 synsets)

- 1 {natural language tongue mother tongue}(678.7 6): the language of a community;
- 2 {weapon arm weapon_system}(673.76 6): used in fighting or hunting
- 3 {head chief top_dog}(671.55 5):
- 4 {capitalist}(669.45 4): a person who believes in the capitalistic system
- 5 {point point_in_time}(669.29 8): a particular clock time;

Table 2: Synsets by flat-probability grouping method

get abstraction level. This expectation is based on the observation that 3000 node granularity is empirically sufficient for describing the translation patterns for selecting the proper target English verb for one Japanese verb(Ikehara,93).

Table 1 shows the average synset node depth and the distribution of synset node depth of WordNet1.4. Table 2 lists the top five noun synsets in the flat probability groupings of 500 and 3000 synsets. "{}" shows synset. The first and the second number in "()" shows the class frequency and the depth of synset respectively.

Level 500 groupings contain a very abstracted level of synsets such as "action", "time_period" and "natural_object". This level seems to be too general for describing the features of objects. In contrast, the level 3000 groupings contains "natural_language", "weapon", "head.chief", and "point.in.time" which seems to be a reasonable basis for feature description.

There is a relatively big depth gap between synsets in the abstracted synset group. Even in the 500 level synset group, there is a two-depth gap. In the 3000 level synset group, there is 4 depth gap between "capitalist" (depth 4) and "point.in.time" (depth 8). The interesting point here is that "point.in.time" seems to be more abstract than "capitalist," intuitively speaking.

The actual synset numbers of each level of synset groups are 518, 1538, and 3001. Thus the flat probability grouping method can precisely control the level of abstraction. Considering the possible abstraction levels available by the flat-depth method, i.e., depth 2 (122 synsets), depth 3 (966 synsets), depth 4 (2949 synsets), this is a great advantage over the flat probability grouping.

5.2 Experiment: Abstracted triples from 1010 corpus

A preliminary experiment for obtaining abstract triples as a basis of features of synsets was conducted. 82,703 surface svo triples are extracted from the 1010 corpus. Polarities of abstracted triple sets for 500, 1500, 3000 level abstraction are 1.20M, 2.03M and 2.30M respectively. Each

Level 500

- 1 {organization}{talk speak utter mouth verbalize verbify}
{organization} (70,4,24,108)
- 2 {organization}{talk speak utter mouth verbalize verbify}
{action} (56,3,35,112)
- 3 {organization}{change undergo.a.change become.different}
{possession} (60,2,83,188)
- 4 {organization}{talk speak utter mouth verbalize verbify}
{event} (48,1,75,84)
- 5 {organization}{move displace make.move}{action}
(50,1,34,82)

Level 3000

- 1 {expert}
{greet recognize}/"express greetings upon meeting ..."
{due.process due.process.of.law}
- 2 {jury}/"a body of citizens sworn to give a true verdict ..."
{pronounce label judge}/"pronounce judgment on"
{capitalist} (4,1,09,4)
- 3 {police police.force constabulary law}
{allege aver say}/"He alleged that we was the victim ..."
{female female.person} (4,1,3)
- 4 {assembly}"a body that holds formal meetings"
{refuse reject pass.up turn.down decline}/"refuse to accept"
{request petition solicitation} (6,0,25,6)
- 5 {animal animate.being beast brute creature fauna}/
{win gain}/"win something through one's efforts"
{contest competition} (5,0,49,6)

"()" shows (# of Surface-Support, Frequency, Occurrences)

Table 3: Example of abstracted triples

abstract triple holds frequency, occurrence number, and word list which supports each of three abstracted synsets.

A filtering heuristic that eliminates abstract triples whose surface support is three (i.e., supported by only one surface pattern) is applied to each set of abstracted triples, and results in the following sizes of abstracted triple sets in the 329K (level 500), 150K (level 1500) and 56K (level 3000) respectively. Each triple is assigned an evaluation score which is a sum of normalized surface support score (= surface support score/maximum surface support score) and normalized frequency (= frequency / maximum frequency). Table 3 shows the top five abstracted triples with respect to their evaluation scores. Items in the table shows subject synset, verb synset, object synset, surface support, frequency and occurrence numbers.

All the subjects in the top five abstract triples of level 500 are "organization". This seems to be reasonable because the contents of the 1010 corpus are news articles and these triples seem to show some highly abstract briefing of the contents of the corpus.

The effectiveness of the filtering and/or scoring heuristics can be measured using two closely related criteria. One measures the plausibility of abstracted triples i.e., the recall and precision ratio of the plausible abstracted triples. The other criteria shows the correctness of the mappings of the surface triple patterns to abstracted triples.

varsity United.Nations team subsidiary State state staff soviet school Politburo police patrol party panel Organization order operation newspaper mission Ministry member magazine line law.firm law land Justice.Department jury industry house headquarters government gang FIA division court country council Conference company committee college club Cabinet business board association Association airline

Table 4. Surface supports of "organization"

This is measured by counting the correct surface supports of each abstracted triple. For example, considering a set of surface words supporting "organization" of the 1 of level 500 shown in table 4, the word "panel" might be used as "panel board". This ability is also measured by developing the word sense disambiguator which inputs the surface triple and selects the most plausible deep triple based on abstracted triple scores matched with the deep triple. The surface supports in Table 4 show the intuitive tendency that a sufficient number of triple data will generate solid results.

6 Conclusions

This paper described a similarity calculation model between objects based on common and distinctive features and proposes an implementation procedure for obtaining features based on abstract triples extracted from a large text corpus (1010 corpus) utilizing taxonomical knowledge (WordNet). The experiment, which used around 80K surface triples, shows that the abstraction level 3000 provides a good basis for feature description. A feature extraction experiment based on large triple data is our next goal.

References

- Kenneth Church and Patrick Hanks. 1989. *Word association norms, mutual information, and lexicography*, In Proceedings of the 27th Annual Meeting of ACL.
- EDR 1995. *Summary for the EDR Electronic Dictionary Version 1 Technical Guide EDR TR2-005*, Japan Electronic Dictionary Research Institute, Tokyo.
- Gregory Grefenstette 1994. *Explorations in Automatic Thesaurus Discovery*, Kluwer Academic Publishers.
- Marti. A. Hearst and Hinrich Schutze. 1993. *Customizing a Lexicon to Better Suit a Computational Task*. Proceedings of the ACL SIGLEX Workshop, Columbus, Ohio.
- Donald Hindle. 1990. *Noun classification from predicate argument structures*, Proceedings of the 28th Annual Meeting of ACL.
- S. Ikehara, M. Miyazaki, A. Yokoo. 1993. *Classification of Language Knowledge for Meaning Analysis in Machine Translation*, Transactions of Information Processing Society of Japan, Vol 34., No.8, pps. 1692-1704.
- J. Kolodner and C. Riesbeck. *Case-Based Reasoning*, tutorial textbook of 11th IJCAI.
- George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, Katherine Miller. 1990. *Five Papers on WordNet*, Cognitive Science Laboratory Report 43, Princeton University.
- Roy Rada, Hafedh Mili, Ellen Bicknell, Maria Blettner. 1989. *Development and Application of a Metric on Semantic Nets*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 1.
- Philip Resnik. 1995a. *Using Information Content to Evaluate Semantic Similarity in a Taxonomy*, Proceedings of IJCAI-95.
- Philip Resnik. 1995b. *Disambiguating Noun Groupings with Respect to WordNet Senses*, Proceedings of Annual Meeting of ACL.
- Hinrich Schutze. 1993. *Advances in Neural Information Processing Systems 5*, Stephen J. Hanson, Jack D. Cowan, G.Lee Files editors, Morgan Kaufmann, San Mateo CA.
- Michael Sussna. 1993. *Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network*, Proceedings of the Second International Conference on Information and Knowledge Management (CIKM-93).
- Amos Tversky. 1977. *Features of Similarity*. Psychological Review, Vol. 84, Number 4.