Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Word sense disambiguation using semantic relatedness measurement

YANG Che-Yu

(*Department of Computer Science and Information Engineering, Tamkang University, Taipei 25137, Taiwan, China*)

E-mail: 890190100@s90.tku.edu.tw

Received Nov. 7, 2005; revision accepted Mar. 17, 2006

Abstract: All human languages have words that can mean different things in different contexts, such words with multiple meanings are potentially "ambiguous". The process of "deciding which of several meanings of a term is intended in a given context" is known as "word sense disambiguation (WSD)". This paper presents a method of WSD that assigns a target word the sense that is most related to the senses of its neighbor words. We explore the use of measures of relatedness between word senses based on a novel hybrid approach. First, we investigate how to "literally" and "regularly" express a "concept". We apply set algebra to WordNet's synsets cooperating with WordNet's word ontology. In this way we establish regular rules for constructing various representations (lexical notations) of a concept using Boolean operators and word forms in various synset(s) defined in WordNet. Then we establish a formal mechanism for quantifying and estimating the semantic relatedness between concepts—we facilitate "concept distribution statistics" to determine the degree of semantic relatedness between two lexically expressed concepts. The experimental results showed good performance on Semcor, a subset of Brown corpus. We observe that measures of semantic relatedness are useful sources of information for WSD.

Key words: Word sense disambiguation (WSD), Semantic relatedness, WordNet, Natural language processing
doi:10.1631/jzus.2006.A1609 **Document code:** A **CLC number:** TP399:H03

INTRODUCTION

The need to determine the degree of semantic similarity, or more generally, relatedness, between two lexically expressed concepts is applied in such applications as word sense disambiguation (WSD), determining discourse structure, text summarization and annotation, information extraction and retrieval, automatic indexing, lexical selection, and automatic correction of word errors in text.

All human languages have words that can mean different things in different contexts, such words with multiple meanings are potentially "ambiguous". For almost all applications of language technology such as machine translation and text retrieval, word sense ambiguity has been a potential source of error. WSD is the process of deciding which of their several meanings is intended in a given context. It has been very difficult to formalize the automatic process of disambiguation. In many ways, WSD is similar to part-of-speech tagging. It involves labelling every word in a text with a tag from a pre-specified set of tag possibilities for each word by using features of the context and other information.

Human beings are especially sophisticated at WSD. For example, given the sentence "The bank holds the mortgage on my home", we immediately know that the bank here refers to a financial institution that accepts deposits and channels the money into lending activities. Whereas given the sentence "He sat on the bank of the river and watched the currents", the bank here means the sloping land beside a body of water. But unfortunately, it is very difficult for computers to do the same job effortlessly. Polysemy—a single word form having more than one meaning; synonymy—multiple words having the same meaning, are both important issues in natural language processing or artificial intelligence related fields.

The research on WSD has been one of the most difficult issues in computational linguistics for a long while. Roughly speaking, recent advances benefit from machine learning techniques, sophisticated sense inventories (especially WordNet) (Fellbaum, 1998), and large corpora to find relevant linguistic features.

Generally, supervised approaches (Yarowsky, 1992; Bruce and Wiebe, 1994; Lin, 1999; Kilgarriff and Rosenzweig, 2000), which learn from correctly sense-annotated corpora, achieve better performance; however, they are highly coordinated to the training corpora, and need large amount of high quality annotated data to achieve reliable results. However, to create large sense annotated corpora is a time-consuming and labor-intensive job, and sometimes the judgment is subjective and thus may be different by individuals. On the other hand, unsupervised methods (Yarowsky, 1992; Agirre and Rigau, 1996; Resnik, 1995; Lin, 2000) have the advantage of making judgment without the need for training and thus can be adopted immediately. However, presently they tend to perform less well than the former.

In this paper, we try to utilize the method of semantic relatedness measurement to accomplish the WSD. We propose the "concept distribution statistics" to quantify and estimate the degree of semantic relatedness between two concepts (synsets) defined in WordNet (Miller, 1995). The approach combines two features: word definition in highly precise thesaurus (WordNet) manually constructed by domain experts; and the actual word usage by numerous ordinary people around the world. The first feature is used for constructing various representations of a concept by appropriate words; and the second feature is used for capturing the distribution of concepts by using statistics of word appearances.

This article is written not only from a theoretical perspective on concept representation, concept distribution and semantic relatedness, but also considered the possible application of the proposed theory on WSD. In the next section, we introduce the online machine-readable semantic dictionary—WordNet. The proposed novel semantic relatedness measuring and WSD method is discussed in Section 3. The experiments and discussions of our WSD methodology are presented in Section 4. The related works on WordNet-based WSD is discussed in Section 5. Then, a conclusion is drawn.

WORDNET—AN ONLINE LEXICAL DATABASE

WordNet is a machine-readable dictionary (MRD) developed by George Miller and his colleagues at the Cognitive Science Laboratory at Princeton University. It is an online lexical database designed for use under program control, and provides a more effective combination of traditional lexicographic information and modern computing (Miller *et al*., 1990). In WordNet a synonym set (synset) represents a single distinct sense or concept. For example, in WordNet, the synset {car, auto, automobile, machine, motorcar} represents the concept of "4-wheeled motor vehicle; usually propelled by an internal combustion engine".

WordNet stores information about words that belong to four parts-of-speech: nouns, verbs, adjectives and adverbs. In WordNet 2.0, there are 152059 words organized in 115424 synsets, approximately 20% of the words in WordNet are polysemous; approximately 40% have one or more synonyms, some 300 prepositions, pronouns, and determiners—although they play an important role in many natural language parsing systems, they are given no semantic illustration in WordNet (Miller *et al*., 1990). Table 1 lists some of the statistics of WordNet 2.0.

Table 1 Number of words, synsets, and word-sense pairs in WordNet 2.0

POS	Unique strings	Synsets	Total word-sense pairs
Noun	114648	79689	141690
Verb	11306	13508	24632
Adjective	21436	18563	31015
Adverb	4669	3664	5808
Total	152059	115424	203145

WordNet database groups English nouns, verbs, adjectives, and adverbs into sets of synonyms that are in turn linked through semantic relations that determine word definitions and senses. WordNet 2.0 features a rich set of 333612 relation links among words, between words and synsets, and between synsets. Table 2 lists some of the semantic relations (links) defined in WordNet (Fellbaum, 1998).

Table 2 Some semantic relations (links) defined in WordNet

Semantic relation	Meaning	Example
Synonymy	X is similar to $f(X)$	homo, man, human being, human
Hypernym	X is a kind of $f(X)$	Apple is a kind of fruit
Hyponym	$f(X)$ is a kind of X	Zebra is a kind of horse
Holonym	X is a part/member of $f(X)$	Wheel is a part of a car
Meronym	X has part/member $f(X)$	Table has part leg
Antonym	$f(X)$ is the opposite of X	Wet is the opposite of dry

The two most typical relations for nouns are hyponymy and hypernymy. These relations connect two synsets if one referred to by another is "is a kind of", or "is a specific example of". That is, if synset A is a kind of synset B, then A is the hyponym of B, and B is the hypernym of A (Fellbaum, 1998). For instance, {car, auto, automobile, machine, motorcar} are the hyponyms of {motor vehicle, automotive vehicle}, and {motor vehicle, automotive vehicle} are their hypernyms. Table 3 lists the major relations defined for nouns and their statistics.

Table 3 Statistics of semantic relations for nouns in WordNet 2.0

Relation's name	Count	Percentage (%)
Hypernym	81857	33.98
Hyponym	81857	33.98
Member holonym	11849	4.92
Part holonym	6883	2.86
Member meronym	11849	4.92
Part meronym	6883	2.86
Derivationally related for	21491	8.92
…	…	…
Total	240883	100.00

More detailed information on semantic relations for verb, adjective and adverb can be referred to (Fellbaum, 1998). We utilize WordNet to accomplish WSD. We will explain these procedures later in this article.

SEMANTIC RELATEDNESS AND WORD SENSE DISAMBIGUATION

In this section, we propose a novel semantic relatedness measuring method (which combines two features), and use it to facilitate WSD. The first feature is the variability of representation of a concept using Boolean operators and various synsets defined in WordNet (lexical notations for a concept). And the other feature is the distribution of concepts (co-occurrence statistics).

We will first review the basic set theory in modern mathematics, and survey the relationship between this set theory and the WordNet's synonym set (synset). Then we proposed the idea of "lexical notations for a concept", which makes it possible to use different sets of word forms (in synsets) to literally express the same concept defined in WordNet using set algebra. And then, we investigate how to utilize the ideas of various lexical concept notations with the distribution statistics of concepts to facilitate the measuring of semantic relatedness between two given concepts, and how to apply it to WSD task.

Synonym set and basic set theory in modern mathematics

"Sets" are one of the most important and fundamental concepts in modern mathematics. A "set" is a well-defined collection of objects considered as a whole. The objects of a set are called "elements" or "members". WordNet is based on the essence of "synonym set", which is a "set" of English terms with the same part-of-speech that can be interchanged in a certain context. For example, {car, auto, automobile, machine, motorcar} form a synset, because they all can be used to refer to the concept: "4-wheeled motor vehicle; usually propelled by an internal combustion engine". Synsets can be related to each other by semantic relations (links). More detail about WordNet can be referred to in Section 2.

In mathematics, there are several ways to construct new sets from existing ones. First, a new set can be constructed by determining which members two sets have "in common", called the "intersection" of sets, as shown in Definition 1.

Definition 1 The "intersection" of two sets A and B is the set that contains all elements of A that also belong to B (or equivalently, all elements of B that also belong to A), but no other elements:

$$A\cap B=\{x|\ x\in A \text{ and } x\in B\}.$$

Let us see what happens when we apply "intersection" to WordNet's synonym sets. Take two synsets {shot, pellet} and {shot, injection} as example. Lexically, the intersection of the two sets is the element "shot", that is, {shot, pellet}∩{shot, injection} ={shot}. But if we view this from the semantic (conceptual) aspect, the intersection of the two sets would be Ø (empty set—the set with no elements), since these two synsets have no synonymy intersection. The idea is shown as Fig.1.

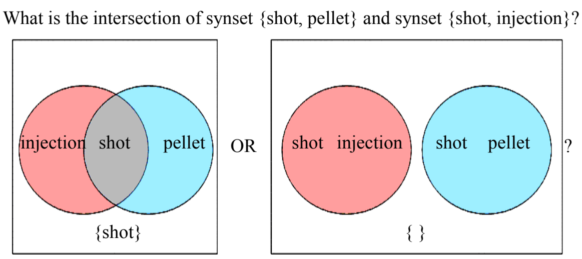


Fig.1 Intersection of WordNet's synonym sets (synsets)

This is due to the fact that the term "shot" is a polysemy word, and it exists in the two sets as two different senses. In the set {shot, injection}, the term "shot" bears the sense "the act of putting a liquid into the body by means of a syringe", for example, "the nurse gave him a flu shot". On the other hand, in the set {shot, pellet}, it carries the sense "a solid missile discharged from a firearm", for example, "the shot buzzed past his ear".

Besides intersection, two sets can also be "added" together to form a new set, called the "union" of the sets, as Definition 2.

Definition 2 If A and B are sets, then the "union" of A and B is the set that contains all elements of A and all elements of B, but no other elements:

$$A\cup B=\{x|\ x\in A \text{ or } x\in B\}.$$

Continue with the last example, according to Definition 2, {shot, injection}∪{shot, pellet}={shot, injection, pellet}. Obviously, the result set is conceptually unclear, since the words "injection" and "pellet" have no common meaning. The idea is illustrated by Fig.2.

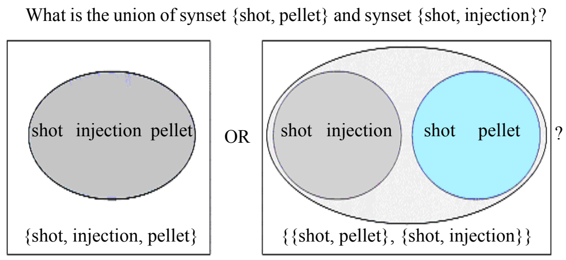


Fig.2 Union of WordNet's synonym sets (synsets)

We clarify the above discussion as Conception 1.

Conception 1 A "synonym set (synset)", although it is a set itself, when applying to the mathematic basic set theory in this research, should be viewed as an impartible/inseparable atomic "element", rather than be treated as a "set".

The idea is reasonable since synset itself is a collection of terms aggregated to represent one single concept of the real world. Any changes to the content (elements) of synsets may result in unpredictable consequence. To treat synsets as atomic elements of set-operations means to manipulate the terms at the semantic level rather than at the lexical level.

Variable lexical notations for a concept

In psychology, a "concept" is a fundamental coding unit in human memory; in WordNet, each synset is constructed to use English terms to represent a single unique concept of the world, as Conception 2.

Conception 2 Each synset in WordNet represents a unique "concept".

Continue with the last example: if we perform "union" operation to two synsets {shot, injection}∪{shot, pellet}, according to Conception 1, the result would be {{shot, injection}, {shot, pellet}}, which is a set of synsets.

In the example, although the meaning of the result set {{shot, injection}, {shot, pellet}} is conceptually unclear, but is it possible that the union of two (or more) other synsets is also a synset? It is just the same to ask: "Is there any set that consists of more than one synsets as its elements and is also a synset itself?", or "Can we group some synsets (concepts) to form another synset (concept)?" Let us see the following definition first.

Definition 3 If every member of the set A is also a member of the set B, then A is said to be a "subset" of B, written $A\subseteq B$.

The fundamental design that lexicographers try to impose on the semantic memory for nouns is not a circle, but a tree. In WordNet, the hyponymy/hypernymy is a semantic relation between word meanings. For instance, {maple} is a hyponym of {tree}, and {tree} is a hyponym of {plant}. According to Table 3, these two relations are in possession of almost 68% in all relations for noun in WordNet. The hyponymy/hypernymy is similar to the subordination/superordination, subset/superset, or the ISA relation in the above discussion. A concept represented by the synset $\{x_1, x_2, \ldots\}$ is said to be a hyponym of the concept represented by the synset $\{y_1, y_2, \ldots\}$ if native speakers of English accept sentences constructed from such frames as "An x is a (kind of) y". The relation can be represented by including in $\{x_1, x_2, \ldots\}$ a pointer to its hypernyms, and including in $\{y_1, y_2, \ldots\}$ pointers to its hyponyms.

If we observe WordNet's hypernym/hyponym relations, a "set" formed by the union of all children synsets/nodes of a father synset/node would also be a synset, which is actually the father synset/node itself. In the same way, these children synsets/nodes are the unions of their own children synsets/nodes; and so on until to the leaf-synsets in the tree. In this circumstance, the union of synsets does result in another synset. Here the leaf-synsets are at the bottom of the WordNet hypernym/hyponym hierarchy, and we treat them as the minimal/atomic "concepts" in the WordNet hierarchy. From the opposite viewpoint, some synsets/nodes group together to form a synset/node that carries a more generic concept. We concluded these as Conception 3.

Conception 3 If we look at hypernym/hyponym relations in the WordNet, the semantic net will become a tree hierarchy. Each synset/node in the hierarchy represents the concept which is formed by the union of concepts of all its children synsets/nodes.

According to (Miller *et al.*, 1990), WordNet partitions the nouns with a set of semantic primes—a (related small) number of generic concepts are selected, and each one of them are treated as the unique beginner of a separate hierarchy. These multiple hierarchies correspond to relatively distinct semantic fields, each with its own vocabulary. In other words, since the features that characterize a unique beginner are inherited by all of its hyponyms, a unique beginner can be regarded as a primitive semantic component of all words in its hierarchically structured semantic field.

According to Conceptions 2 and 3, there emerges a dawn that: when we look at the hypernymy/hyponymy relations, some semantically related synsets can be transformed between/into each other yet keeping the represented "concept" constant. Now we investigate how to use algebra of set to represent a concept/synset using some other more specific or more general concepts/synsets in the WordNet hierarchy, yet keeping the original meaning and scope unchanged.

1. Generic concept notation for a synset

In Fig.3, according to Conception 1 and Conception 3, we have:

$$\begin{aligned}
&D = I \cup J \cup K \ \text{(Here } D, I, J, K \text{ are synsets)}\\
&\Rightarrow J = D-(I\cup K) = (D-I)\cap(D-K)\\
&\quad = D\cap(\overline{I\cup K}) = D\cap(\overline{I}\cap\overline{K})\\
&\qquad \text{(1-level generic notation)} \qquad (1)
\end{aligned}$$

$$\begin{aligned}
&B = D\cup E\cup F\\
&\Rightarrow D = B-(E\cup F) = (B-E)\cap(B-F)\\
&\quad = B\cap(\overline{E\cup F}) = B\cap(\overline{E}\cap\overline{F}) \qquad (2)
\end{aligned}$$

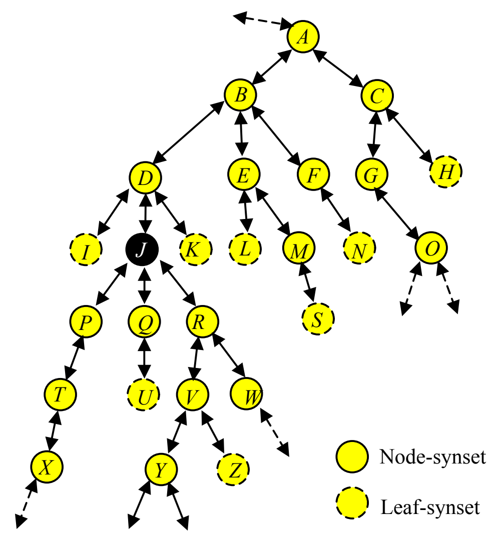


Fig.3 An example "snapshot" of WordNet hypernym/hyponym hierarchy (the nodes are synsets)

By replacing D in Eq.(1) with that in Eq.(2), we get:

$$J = D\cap(\overline{I}\cap\overline{K}) = (B\cap(\overline{E}\cap\overline{F}))\cap(\overline{I}\cap\overline{K})$$
$$= B\cap((\overline{E}\cap\overline{F})\cap(\overline{I}\cap\overline{K}))$$
(2-level generic notation) (3)

Now let us apply some actual data found in WordNet into this example. In WordNet, we found the following instance:

(D) fisherman's lure, fish lure—(angling) any bright artificial bait consisting of plastic or metal mounted with hooks and trimmed with feathers

@→(I) spinner—fisherman's lure; revolves when drawn through the water;

@→(J) fly—fisherman's lure consisting of a fishhook decorated to look like an insect;

@→(K) troll—a fisherman's lure used in trolling; "he used a spinner as his troll".

(Note: here "@→" represents the "hyponymy" semantic relation)

{fisherman's lure, fish lure}, {spinner}, {fly} and {troll} are synsets. Apply these data into Eq.(1), the 1-level generic notation for the synset {fly} can be expressed by:

$$\{\text{fisherman's lure, fish lure}\}\cap\left(\overline{\{\text{spinner}\}}\cap\overline{\{\text{troll}\}}\right). \quad (4)$$

For better clearness and to prevent confusion, when we apply the lexical data (word forms) of synsets into algebra operation, we use the Boolean operators "AND", "OR", "NOT" and parenthesis instead of the notation of "$\cap$", "$\cup$", "$\overline{\ }$" and brace respectively.

Thus, the 1-level generic notation for the synset ("fly") becomes:

("fisherman's lure" OR "fish lure") AND ((NOT "spinner") AND (NOT "troll")). (5)

For the example of 2-level generic notation, we find the following hypernymy data in WordNet:

(B) bait, decoy, lure—something used to lure victims into danger

@→(D) fisherman's lure, fish lure—(angling) any bright artificial bait consisting of plastic or metal mounted with hooks and trimmed with feathers;

@→(E) ground bait—bait scattered on the water to attract fish;

@→(F) stool pigeon—a dummy pigeon used to decoy others.

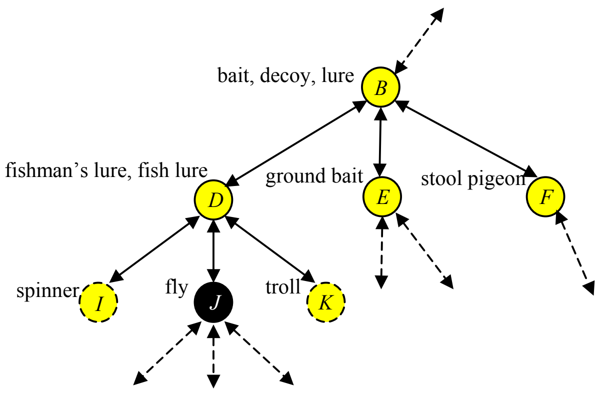
Apply this and the above data into Eq.(3), the 2-level generic notation for the synset ("fly") would be:

$$\{\text{bait, decoy, lure}\}\cap\left(\left(\overline{\{\text{ground bait}\}}\cap\overline{\{\text{stool pigeon}\}}\right)\cap\left(\overline{\{\text{spinner}\}}\cap\overline{\{\text{troll}\}}\right)\right). \quad (6)$$

Or in another form as:

("bait" OR "decoy" OR "lure") AND (((NOT "ground bait") AND (NOT "stool pigeon")) AND ((NOT "spinner") AND (NOT "troll"))). (7)

This example is summarized as Fig.4.



1-level generic notation of the synset J("fly")

$J = D\cap(\overline{I}\cap\overline{J})\Rightarrow$

("fly")=("fisherman's lure" OR "fish lure") AND ((NOT "spinner") AND (NOT "troll"))

2-level generic notation of the synset J("fly")

$J = B\cap((\overline{E}\cap\overline{F})\cap(\overline{I}\cap\overline{K}))\Rightarrow$

("fly")=("bait" OR "decoy" OR "lure") AND (((NOT "ground bait") AND (NOT "stool pigeon")) AND ((NOT "spinner") AND (NOT "troll")))

Fig.4 Examples of more generic concept notation for a synset

We normalize the i-level generic notation for a synset as Definition 4.

Definition 4 (i-level generic notation for a synset) If S_k is a synset, F_i is the synset that is located i link(s) away following the hypernym links from S_k, then the i-level generic notation for S_k is:

$$GE_i(S_k) = F_i \bigcap_{m=1}^{i} \Big(\bigcap_{S_j \in F_m, j \neq k} \overline{S_j} \Big), \tag{8}$$

where F_i is the parent node of F_{i-1}, F_{i-1} is the parent node of $F_{i-2}\ldots$

Besides using more generic concepts to represent a concept, we can also do the opposite. The following is to use more specific concepts to represent a concept.

2. Specific concept notation for a synset

In Fig.3, according to Conception 1 and Conception 3, we have:

$$J = P \cup Q \cup R \text{ (Here } J, P, Q \text{ and } R \text{ are synsets)}$$
$$\text{(1-level specific regular notation)} \tag{9}$$

$P = T$, $Q = U$, $R = V \cup W$, by replacing P, Q, R in Eq.(9) with these, we get:

$$J = P \cup Q \cup R = T \cup U \cup (V \cup W)$$
$$\text{(2-level specific regular notation)} \tag{10}$$

$P = P \cup T$, $Q = Q \cup U$, $R = R \cup V \cup W$, by replacing P, Q, R in Eq.(9) with these, we get:

$$\begin{aligned} J &= P \cup Q \cup R \\ &= (P \cup T) \cup (Q \cup U) \cup (R \cup V \cup W) \\ &= (P \cup Q \cup R) \cup (T \cup U \cup V \cup W) \end{aligned}$$
$$\text{(2-level specific extended notation)} \tag{11}$$

Here we have two kinds of specific notation for a synset, one is "regular notation" and the other is "extended notation". The extended notation differs from the regular notation in that the extended notation includes all the nodes on the paths to the lower level in the hierarchy. This makes sense when we apply the lexical information (word forms) of synsets into the notations—to use more word forms to represent a concept.

We normalize the "*i*-level specific regular notation" and the "*i*-level specific extended notation" for a synset as Definition 5 and Definition 6 respectively.

Definition 5 (*i*-level specific regular notation for a synset) If S is a synset, L_i is the set of synsets C_{ik} that are located i link(s) away following the hyponym links from S, then the *i*-level specific regular notation for S is:

$$SE_i(S) = L_i = \bigcup_{C_{ik} \in L_i} C_{ik}, \tag{12}$$

where if C_{ik} is null, then $C_{(i-1)k}$ would be used ($C_{(i-1)k}$ is a leaf node in the case).

Definition 6 (*i*-level specific extended notation for a synset) If S is a synset, L_i is the set of synsets C_{ik} that are located i link(s) away following the hyponym links from S, then the *i*-level specific extended notation for S is:

$$SE_i(S) = \bigcup_{j=0}^{i} L_j = \bigcup_{j=0}^{i} \left(\bigcup_{C_{jk} \in L_j} C_{jk} \right), \tag{13}$$

where if C_{ik} is null, then C_{ik} would be ignored ($C_{(i-1)k}$ is a leaf node here).

Semantic relatedness computing and word sense disambiguation

The proposed WSD method is a hybrid approach that combines a knowledge-rich ontology—WordNet, with knowledge-poor corpus statistics—the Internet (World Wide Web pages).

Here we establish a formal mechanism for quantifying and estimating the semantic relatedness between concepts—we facilitate "concept distribution statistics" to determine the degree of semantic relatedness between two lexically expressed concepts defined in WordNet.

The insight of our semantic relatedness measurement method is to observe the "distribution of concepts". The major issue is—what is "related" and what is "unrelated"? Or what is "more related" and what is "less related"? We take both the "frequency" part and the "geographical" part of concept distribution into consideration, and define the relatedness as Conception 4.

Conception 4 Concepts that appear more frequently and closer with each other are "more related" to each other than the concepts that appear less frequently and farther are.

Our WSD model is just derived from Conception 4. To implement and make Conception 4 computable, we define the mappings in Table 4.

According to Table 4, Conception 5 is derived.

Conception 5 For any pair of WordNet synsets, the more web pages contain both synsets' concepts (lexical notations), the more semantic relatedness the synsets have.

Table 4 Mapping from Conception 5 to the semantic relatedness measurement

Conception 4	Semantic relatedness measurement
Concepts	WordNet synsets (their generic/specific lexical notations)
Geographical distribution (close or far of appearance)	Existing (be referred to) in a web page or not (Boolean)
Frequency distribution (co-occurrence frequency)	Number of web pages that contain the synsets

Not only the World Wide Web is the most rich and domain extensive natural language text resource, but also the content is very up to date and grows continuously. As the context of web pages on the Internet are composed by innumerable people, the validity for use as a WSD source is thus guaranteed. Furthermore, there are now a few sophisticated and powerful keyword-based (lexical) World Wide Web search engines that can be used to help the task.

So based on Definitions 4, 5, 6 (generic/specific notation for a synset) and Conception 5, the procedure for determining the semantic relatedness of two given WordNet synsets is as shown in Fig.5.

Now we try to use the procedure for determining the semantic relatedness of two given WordNet synsets to facilitate the WSD. The WSD task on textual data against WordNet is shown in Fig.6. The WSD task here is actually the task of generating the mapping between the word forms in a text and the word sense inventory in WordNet.

The proposed WSD procedure for a term is briefly described as follows: to disambiguate a term (called a "target term"), the target term and a nearby context term are paired (excluding stop words), and a "score" of relatedness is calculated by searching the Internet with queries formed by using different senses (synsets or their generic/specific notation) of the two terms. The notations of the two concepts are concatenated by the Boolean operator "AND" to find web pages that contain both of them (recall Conception 5). If one term has m senses and the other term has n senses, then there will be $m\times n$ sense-pairs to be compared. To gather more evidence for disambiguation, similar process can be carried out several times for a target term with more than one context terms, then sum up the results scores of each sense. Each sense is then ranked according to the total score it gets. In this way the most possible sense for the target term would be the one with the highest score. The procedure is illustrated by Fig.7.

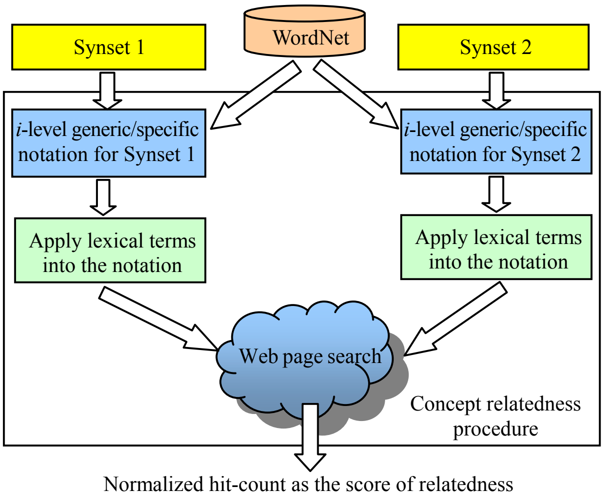


Fig.5 Procedure for determining the semantic relatedness of two given WordNet synsets

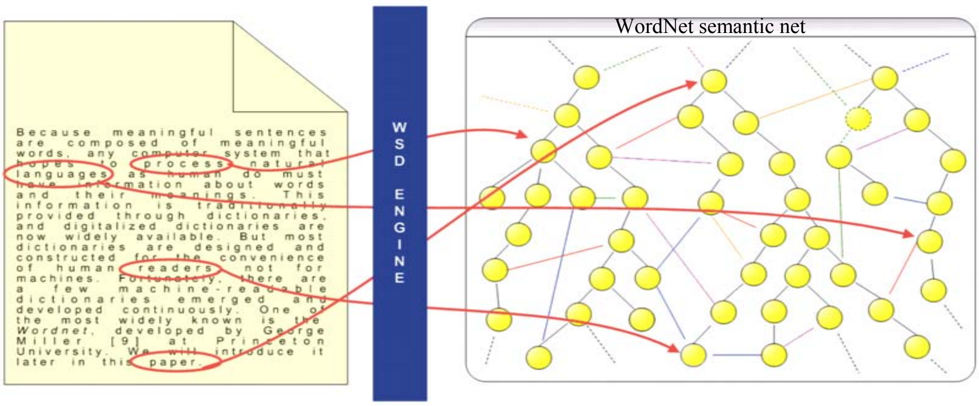


Fig.6 The WSD task using WordNet—to generate the mappings between word forms and synsets

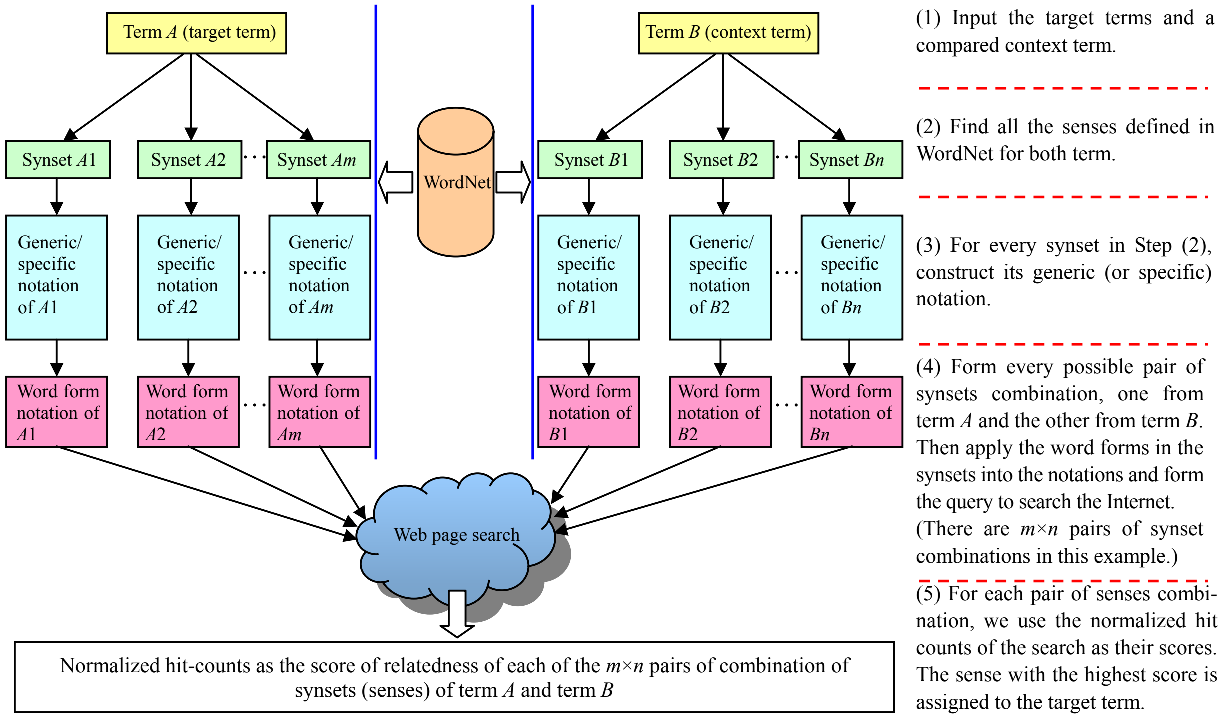


Fig.7 The procedure to find the most appropriate sense for a term

EVALUATION

The experiment was tested on ALL the nouns in a subset of an unrestricted public domain corpus, thus has to make fine-grained distinctions among all the senses in WordNet.

Experimental setup

We test our WSD procedure on the SemCor corpus. SemCor corpus, created by Princeton University, is a subset of the English Brown Corpus containing about 700000 running words. In SemCor all the words are tagged by part of speech (POS), and more than 200000 content words are also lemmatized and sense-tagged manually according to Princeton WordNet such as the example in Fig.8. More in detail, the SemCor corpus is composed of about 352 texts. The “all-words” component of SemCor has about 359732 tokens among which almost 192639 are semantically annotated.

Semcor inventories various genders of text, both informative and imaginative prose. We randomly selected four text files from SemCor: br-a01 (Press Reportage), br-b20 (Press Editorial), br-j09 (Learned) and br-r05 (Humor) to test with. In the four texts, the total number of target noun words to be disambiguated is around 13225, the average polysemy degree (number of senses per term) of these target noun words is 4.08. Table 5 lists the polysemy statistical information of the four texts.

Any Internet search engine that supports AND, OR, NOT, and parenthesis operators can be utilized to obtain concept co-occurrence statistics in our experiment. However, some adjustments need to be made to the concept notations in Section 3.2 according to the searching syntax usages of individual search engine. Google (http://www.google.com) offers public Web APIs service to software developers, however, it has a limitation of a maximum number of words in the query of 10 words, thus we choose to use Yahoo (http://tw.yahoo.com).

First, a context window consists of a target word and some number of surrounding context words is determined, and a set of candidate senses is identified for each word in the window based on the sense inventory in the WordNet. Then a $Score_k$ will be assigned to each possible sense k of the target word, computed by adding together the relatedness scores

Table 5 Polysemy statistical information of the four tested text files

Polysemy degree	2	3	4	5	6	7	8	9	10	11<	Total
Instances	2142	3809	1597	1190	1106	949	492	386	537	1017	13225
Percentage (%)	16.2	28.8	12.1	9.0	8.4	7.2	3.7	2.9	4.1	7.7	100

Nothing in English has been ridiculed as much as the ambiguous use of words, unless it be the ambiguous use of sentences.

```
<contextfile concordance=brown>
<context filename=br-r05 paras=yes>
<p pnum=1>
<s snum=1>
<wf cmd=done pos=NN lemma=nothing wnsn=1
    lexsn=1:23:00::>Nothing</wf>
<wf cmd=ignore pos=IN>in</wf>
<wf cmd=done pos=NN lemma=english wnsn=1
    lexsn=1:10:00::>English</wf>
<wf cmd=done pos=VBZ ot=notag>has</wf>
<wf cmd=done pos=VBN ot=notag>been</wf>
<wf cmd=done pos=VB lemma=ridicule wnsn=1
    lexsn=2:32:00::>ridiculed</wf>
<wf cmd=done pos=RB ot=complexprep>as_much_as</wf>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=JJ lemma=ambiguous wnsn=1
    lexsn=3:00:04::>ambiguous</wf>
<wf cmd=done pos=NN lemma=use wnsn=1
    lexsn=1:04:00::>use</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done pos=NN lemma=word wnsn=1
    lexsn=1:10:00::>words</wf>
<punc>,</punc>
<wf cmd=ignore pos=IN>unless</wf>
<wf cmd=ignore pos=PRP>it</wf>
<wf cmd=done pos=VB lemma=be wnsn=2
    lexsn=2:42:06::>be</wf>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=JJ lemma=ambiguous wnsn=1
    lexsn=3:00:04::>ambiguous</wf>
<wf cmd=done pos=NN lemma=use wnsn=1
    lexsn=1:04:00::>use</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done pos=NN lemma=sentence wnsn=1
    lexsn=1:10:00::>sentences</wf>
<punc>.</punc>
</s>
…
```

Fig.8 Example sentences and the tag format in Semcor

obtained by comparing the senses of the target word with every sense of every non-target word in the context window.

The sense with the highest *Score* is judged to be the most appropriate sense for the target word. Given the answers generated by the algorithm, we compare them with the human decided answers and compute the precision.

For the value "*i*" in the generic and specific concept notations, values "1", "2" and "3" are set up, which will include one, two and three levels of generic or specific synsets that are found through hypernym or hyponym links from the original synset respectively. For the context window, the sizes of "3", "5", "7" are selected, which means the target word will be compared with two, four and six neighboring context words respectively.

In the experiments, we made some modifications to the generic notation. In Eq.(8), if the value of "*i*" is larger than 1, then the father synsets that are located within $i-1$ links away are not considered in the notations. For example, in Fig.4, the "2-level generic notation of synset J'" does not take synset D into consideration. In the experiment, for better statistical information gathering, we modified it to include these father synsets by concatenating them with "union" operators to the original synset. And the original synset itself is included in the generic notation also. Thus for the example in Fig.4, the "2-level generic notation of synset J'" would become $(J\cup B\cup D)\cap((\overline{E}\cap\overline{F})\cap(\overline{I}\cap\overline{K}))$.

In order to make the generic and the specific notations more comparable with each other, we tested only specific extended notation of Eq.(13), and left specific regular notation behind. However, we believe that the performance of specific regular and specific extended notations should not be far away.

Experimental results

The "baseline" precision is the average possibility of hitting the correct senses for the target terms by wild guess. For example, the word form "eagle" has four different senses, so the baseline precision of this word is 1/4 (25%). Thus the baseline precision values in the experiments are calculated by:

$$\text{Baseline}=1/(\text{average number of senses of all the target terms to be disambiguated}). \quad (14)$$

Tables 6~9 list the WSD results with generic notation on the four randomly selected texts from

Semcor. The data in the cell of the table is the average WSD precision, which is the percentage of correctly disambiguated terms against all the disambiguated terms for each distinct experiment configuration. Each column of the tables represents the *i* value of the *i*-level generic notation, the values of "1", "2" and "3" are set up for the *i*. Each row represents the size of the context window, the values of "3", "5" and "7" are set up for it.

Fig.9 shows the average precision of WSD on the four texts against different *i* values of *i*-level generic notation. Fig.10 shows the average precision of WSD on the four texts against different sizes of context window.

Tables 10~13 list the WSD results with specific notation on the four randomly selected texts from Semcor.

Fig.11 shows the average precision of WSD on the four texts against different *i* values of *i*-level specific notation. Fig.12 shows the average precision of WSD on the four texts against different sizes of context window.

Fig.13 shows the average precision of WSD when the target term has various degree of polysemy. Naturally, the greater degree of polysemy, the harder the task of correct disambiguation.

Discussion

One reason we cannot directly use only the word forms in the target synset to represent a concept can be found in the following example. There are four senses defined for the term "eagle" in the WordNet:

The noun "eagle" has 4 senses:

Table 6 Precision on br-a01 (generic notation) (%)

	1	2	3	Avg.	Baseline
3	76.52	79.03	77.93	77.83	27.16
5	77.70	76.78	75.00	76.49	27.16
7	76.65	76.73	75.97	76.45	27.16
Avg.	76.96	77.51	76.30	76.92	27.16
Baseline	27.16	27.16	27.16	27.16	

Table 7 Precision on br-b20 (generic notation) (%)

	1	2	3	Avg.	Baseline
3	77.55	72.97	75.12	75.21	25.88
5	79.01	74.38	75.87	76.42	25.88
7	78.70	72.21	73.87	74.93	25.88
Avg.	78.42	73.19	74.95	75.52	25.88
Baseline	25.88	25.88	25.88	25.88	

Table 8 Precision on br-j09 (generic notation) (%)

	1	2	3	Avg.	Baseline
3	75.76	75.76	75.76	75.76	22.23
5	82.45	82.45	82.53	82.48	22.23
7	74.82	74.91	74.73	74.82	22.23
Avg.	77.68	77.71	77.67	77.69	22.23
Baseline	22.23	22.23	22.23	22.23	

Table 9 Precision on br-r05 (generic notation) (%)

	1	2	3	Avg.	Baseline
3	73.08	72.14	74.89	73.37	22.74
5	73.73	70.30	73.68	72.57	22.74
7	72.65	70.09	74.12	72.29	22.74
Avg.	73.15	70.84	74.23	72.74	22.74
Baseline	22.74	22.74	22.74	22.74	

Table 10 Precision on br-a01 (specific notation) (%)

	1	2	3	Avg.	Baseline
3	80.37	79.31	75.12	78.27	27.16
5	84.56	81.85	74.40	80.27	27.16
7	76.24	75.39	72.25	74.63	27.16
Avg.	80.39	78.85	73.92	77.72	27.16
Baseline	27.16	27.16	27.16	27.16	

Table 11 Precision on br-b20 (specific notation) (%)

	1	2	3	Avg.	Baseline
3	81.78	73.12	72.50	76.50	25.88
5	85.20	75.01	74.33	79.68	25.88
7	82.15	70.50	71.75	76.45	25.88
Avg.	83.04	76.73	72.86	77.54	25.88
Baseline	25.88	25.88	25.88	25.88	

Table 12 Precision on br-j09 (specific notation) (%)

	1	2	3	Avg.	Baseline
3	77.10	75.50	73.56	75.39	22.23
5	84.75	80.48	76.72	80.65	22.23
7	75.37	73.25	73.25	73.96	22.23
Avg.	79.07	76.41	74.51	76.66	22.23
Baseline	22.23	22.23	22.23	22.23	

Table 13 Precision on br-r05 (specific notation) (%)

	1	2	3	Avg.	Baseline
3	75.13	71.38	72.10	71.39	22.74
5	78.27	72.50	71.53	72.70	22.74
7	74.90	70.22	70.66	70.31	22.74
Avg.	76.10	71.37	66.93	71.47	22.74
Baseline	22.74	22.74	22.74	22.74	

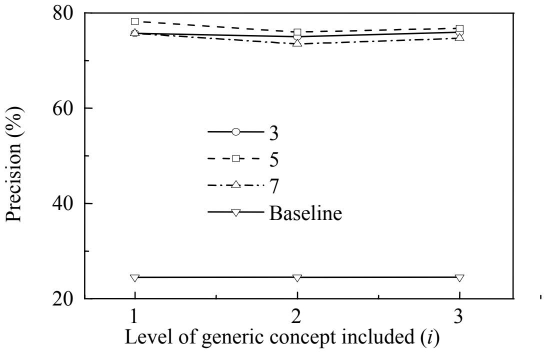


Fig.9 Average precision against different *i* values of the generic notation

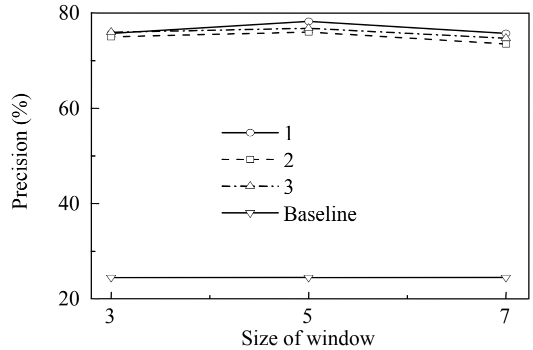


Fig.10 Average precision (generic notation) against different sizes of context window

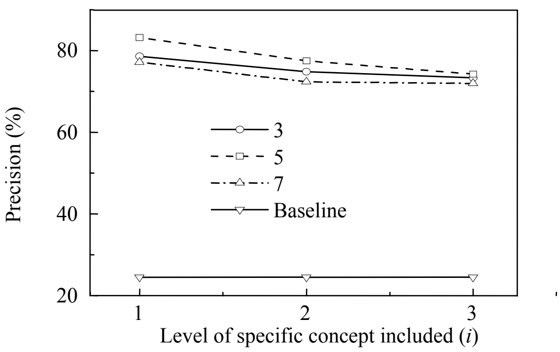


Fig.11 Average precision against different *i* values of the specific notation

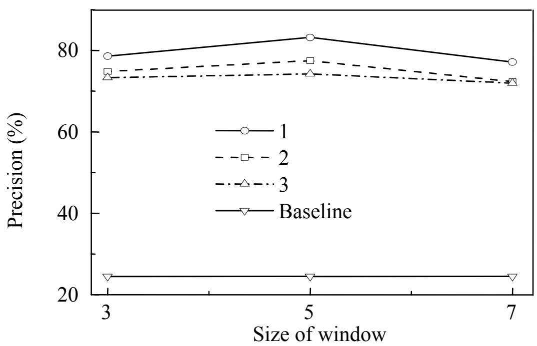


Fig.12 Average precision (specific notation) against different sizes of context window

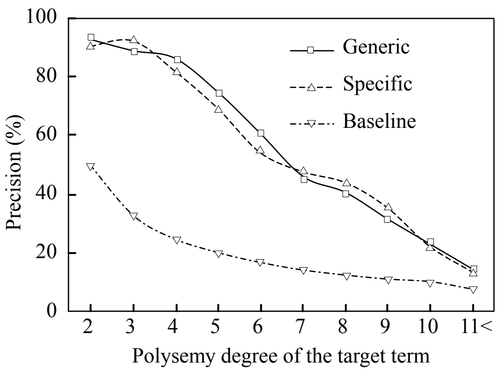


Fig.13 Precision against the degree of polysemy

(1) eagle, bird of Jove—any of various large keen-sighted diurnal birds of prey noted for their broad wings and strong soaring flight;

(2) eagle—(golf) a score of two strokes under par on a hole;

(3) eagle—a former gold coin in the United States worth 10 dollars;

(4) eagle—an emblem representing power, e.g., "the Roman eagle".

In WordNet, sometimes there is only one single word form without other synonym word form(s) in a synset. In WordNet 2.0, the mean size of noun synsets (average number of word forms per synset) is 1.778 (Yu *et al*., 2004). In the above example, only the first sense of the four senses of the word form "eagle" has two word forms in it ("eagle" and "bird of Jove"). If we directly use only the word forms in the target and compared synsets to represent themselves and to construct the Internet search query (without their hypernym/hyponym synsets), then the query string and the search results (hit counts) formed by Senses 2, 3, 4 in this example would be identical. This must be prevented from happening. If we co-operate the synsets with their own generic/hypernym (or specific/hyponym) concepts, we would have the following additional information:

(1) Sense 1

eagle, bird of Jove—various large keen-sighted diurnal birds of prey noted for their broad wings and strong soaring flight.

@→bird of prey, raptor, raptorial bird—any of numerous carnivorous birds that hunt and kill other animals;

@→bird—warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings.

(2) Sense 2

eagle—(golf) a score of two strokes under par on a hole.

@→score—a number that expresses the accomplishment of a team or an individual in a game or contest;

@→number—a concept of quantity derived from zero and units.

(3) Sense 3

eagle—a former gold coin in the United States worth 10 dollars.

@→coin—a metal piece (usually a disc) used as money;

@→coinage, mintage, specie, metal money —coins collectively.

(4) Sense 4

eagle—an emblem representing power, e.g., "the Roman eagle".

@→emblem, allegory—a visible symbol representing an abstract idea;

@→symbol, symbolization, symbolisation, symbolic representation.

Thus, using generic or specific notations of synsets can not only facilitate the construction of distinct literal representations of concepts, but also obtain more evidence for disambiguation.

On the other hand, WordNet makes a great number of fine-grained word sense distinctions. However, a great number of sense distinctions make the problem of WSD harder. The basic idea behind this is as follows. WordNet contains many polysemous word forms. In many cases the level of polysemy is rather high. For instance, around 3000 nouns have 3 senses or more. These sense distinctions are often very fine-grained and show a considerable degree of relatedness. Although WordNet is used as a resource for semantic information in many NLP applications, the sense distinctions in WordNet are too fine-grained for a number of NLP tasks (Kilgarriff, 1997; Slator *et al.*, 1987). Our proposed generic/specific notations for a synset are somewhat like the combinations of the related fine-grained WordNet sense distinctions into larger clusters—to compute the relatedness between groups of synsets instead of between two synsets. This attempts to allow computing the relatedness of synsets at a more coarse-grained level of sense distinction. For example, in Fig.4, the "2-level generic notation of synset *J*". In the experiment, for better statistical information gathering, we modified it to include these father synsets, thus the "2-level generic notation of synset *J*" became $(J \cup B \cup D) \cap ((\overline{E} \cap \overline{F}) \cap (\overline{I} \cap \overline{K}))$. However, to make even more coarse-grained level sense distinction, the "exclusion" may be removed from the notation, so that the notation of J becomes $J \cup D \cup B$.

It is of some interest that there are some words that are not part of the everyday vocabulary from the different levels of the concept hierarchy. For example, look at the three adjacent hypernym/hyponym synsets: {"pony"}, {"horse", "Equus caballus"} and {"horse", "Equus caballus"}. For the synset {"pony"}, it may benefit from using its 1-level generic notation which includes its hypernym synset {"horse", "Equus caballus"}; whereas for the synset {"horse", "Equus caballus"}, it may NOT benefit from including its hypernym synset {"equine", "equid"}. In the same way, it does NOT make lots of difference for the synset {"pony"} to be represented by its 1-level, or 2-level or 3-level generic notation.

On the other hand, when trying to construct the various notations of a synset, the reason we considered only hypernym/hyponym semantic relations and left other relations such as holonym (member/part of), antonym (opposite of), etc., is that hypernym/hyponym is the most uniform relations among the relations defined in WordNet. Almost every synset has a hypernym/hyponym connection to other synsets. This makes it possible to construct the various notations in a consistent manner.

About the size of context window, what is the optimum context-size for disambiguating using our WSD method is an important issue. People could assume that the more context word being compared, the better the disambiguation results would be. Nevertheless the actual nature of each text is for sure an important factor that is difficult to measure. How far the useful context words usually appear around the target to-be-disambiguated word is unsteady, depending on the properties of the different text sources. Thus the best window size depends on the natural property of the text sources.

About the efficiency of the system, the efficiency mainly depends on: the Internet traffic, search engine's response time, the number of surrounding context words being compared and the number of senses the compared words bear. There are some possible approaches to improve the efficiency if it is critical to some circumstance such as online applications. As to the Internet traffic issue, if the computing can be carried out at the same place with the search engine—either to have a local web page index database, or move the computation to the local side of the search engine, the cost of network traffic will be significantly reduced. As to the issue of number of senses the compared words bear, one approach is to abandon the generating and comparing with the concept notations of the compared words, and only directly compute with the compared words themselves (the word forms). It means to disregard which senses the compared words bear in the instance, and directly use the compared word to form the search queries. However, the performance needs to be re-tested for this approach. Another idea is to "cache" the results previously achieved; however, this involves other techniques which are not adopted in our original WSD approach.

Side effect of polysemy

We use the Internet search hit counts to capture the frequency and geographical distribution of concepts. Because the Internet search is based on the lexical match between words in user's query and words in target objects, it happens that—although we use some combination of terms to represent a concept, and use the notations of two concepts as a search query to search the Internet, however, some matches are literally matched but conceptually mismatched, so are false matches for concept distribution statistics. Actually, this happens due to the essence of polysemy of English terms, so we called it "the side effect of polysemy", which produces noise and affects the accuracy of our WSD method. In WordNet 2.0, the average polysemy degree of noun including monosemous words is 1.23 (The average polysemy degree excluding monosemous words is 2.79). It means that in average a noun word has 1.23 different senses in WordNet inventory.

We cannot completely avoid the side effect of polysemy, which is the essence of English terms, however we can reduce its influence on our WSD task. Because we have several different notations (i-level generic or specific) which can be adopted alternately for representing a concept, thus we can choose the one that has the least mean number of senses per term to possibly minimize the chance of the side effect happening. The mean number of senses per term (MSPT) for a notation e is computed by

$$MSPT(e) = \sum_{i=1}^{N} |t_i| \Big/ N, \tag{15}$$

where N is the number of terms in the notation e, t_i is a term in e, $|t_i|$ is the number of senses term t_i bears.

The larger value the MSPT is, the more false matches the Internet search may produce, thus the more serious the side effect of polysemy is. On the contrary, the smaller value the MSPT is, the less false matches the Internet search produces, thus the slighter the side effect of polysemy is. When calculating the relatedness of two given concepts, to enable the choosing of least-MSPT notation to form the Internet search query, we need to allow the different kinds of notations to be chosen for the two compared concepts. For example, one concept adopts its 2-level generic notation, while the other concept adopts its 1-level specific notation, then these two least-MSPT notations are associated to form an Internet search query for acquiring distribution statistics.

RELATED WORKS

The study of semantic relatedness has been a part of artificial intelligence and psychology for many years. Much of the early semantic relatedness study in natural language processing centered around the use of Roget's thesaurus (Yarowsky, 1992). As WordNet became available later, most of the new work utilized it (Agirre and Rigau, 1996; Resnik, 1995; Jiang and Conrath, 1997). These methods treat WordNet or similar resource as both the source of the sense inventory as well as a repository of information about words that can be exploited to distinguish their meanings in text.

In the line of the edge-based approach, semantic distance is calculated using the edge counting principle. If all the edges (branches of the tree) are of equal length, then the number of intervening edges between two synsets is a measure of the distance. The measurement usually used (Rada *et al*., 1989; Lee *et al*., 1993) is the shortest path between concepts. This relies on an ideal taxonomy with edges of equal length. Unfortunately in taxonomies based on natural languages, the edges are not the same length. A number of different methods related to distance using edges have been modified to try to correct the problem of this non-uniformity. These modifications include the density of the sub-hierarchies (Agirre and Rigau, 1996), the depth in the hierarchy where the word is found (Sussna, 1993; Leacock and Martin, 1998), and the type of links (Hirst and St-Onge, 1998). Rosso *et al*.(2003) proposed a method for the resolution of lexical ambiguity relying on the use of the wide-coverage noun taxonomy of WordNet and the notion of conceptual distance among concepts, captured by a conceptual density formula developed for this purpose.

And some hybrid approaches, that combine a “knowledge-rich” source, such as a thesaurus, with a “knowledge-poor” source, such as corpus statistics (Resnik, 1995; Lin, 1998; Jiang and Conrath, 1997), were proposed. Yang *et al*.(2005) overcame the problems of word ambiguity by indexing textual information with its underlying concepts using WordNet and the proposed WSD method (our previous work). A system designed for automatically answering student questions in an e-learning environment was designed. Moldovan and Mihalcea (2000) described a method to disambiguate word senses using the WordNet and the Internet to measure semantic densities between pairs of words. They first determine the most common sense-pairs. This is done only for verb-noun, adjective-noun and adverb-verb pairs. Subsequently, verb-noun pairs are disambiguated by taking the first possible senses of the words (as ranked by the initial algorithm) and calculating “conceptual density” of the pairs by examining the WordNet glosses of the sub-hierarchies. This then ranks each pair of senses by looking at the noun-context of the verb and comparing it with the given noun (and its sub-hierarchy).

Banerjee and Pedersen (2002) began this line of research by adapting the Lesk algorithm (Lesk, 1986) for WSD to WordNet. Lesk’s algorithm disambiguates a target word by selecting the sense whose dictionary gloss shares the largest number of words with the glosses of neighboring words. Gloss overlaps can be viewed as a measure of semantic relatedness. Patwardhan *et al*.(2003) observed that disambiguation can be carried out using any measure that can score the relatedness between two word senses.

Chua and Kulathuramaiyer (2004) explored the notion of semantic feature selection by employing WordNet. The proposed semantic approach employs noun synonyms and word senses for feature selection to select terms that are semantically representative of a category of documents. Information retrieval using word senses is emerging as a good research challenge on semantic information retrieval. Kim *et al*.(2004) proposed a method using word senses in information retrieval: root sense tagging method. For other applications of language technology, WSD is also used for machine translation. Li and Li (2004) proposed a method for word translation disambiguation. They used a machine-learning technique called bilingual bootstrapping.

CONCLUSION

This article is written not only from a theoretical perspective on concept representation, concept distribution and semantic relatedness, but also considered the possible application of the proposed theory on word sense disambiguation, which is in the field of artificial intelligence and natural language processing. In this work we propose a novel semantic relatedness measuring method which is a hybrid approach that combines a knowledge-rich source (WordNet) for word ontology, with a knowledge-poor source (the Internet search) for word distribution statistics.

First, we investigate how to “literally” and “regularly” express a “concept”. We apply set algebra to WordNet’s synsets cooperating with WordNet’s word ontology. Through this we establish regular rules for constructing various representations (lexical notations) of a concept using Boolean operators and word forms in various synset(s) defined in WordNet. Then we establish a formal mechanism for quantifying and estimating the semantic relatedness between concepts. We combine the idea of concept notation with the concept distribution to compute semantic relatedness between two given concepts.

Then we present a method of word sense disambiguation that assigns a target word the sense that is most related to the senses of its neighbor words. We carry out disambiguation relative to the senses defined in the lexical database WordNet. Our method is unsupervised, and does not require any training in advance. The experimental results showed good performance on SemCor. The algorithm can be used with any measure that computes a relatedness score between two concepts. This work has shown that the measurement of semantic relatedness is a feasible approach to word sense disambiguation.

FUTURE RESEARCH

This paper proposed a novel method for word sense disambiguation. An interesting application of it is to facilitate text retrieval, such as question answering (QA) or search engine. Some methods of QA use keyword-based techniques to locate interesting passages and sentences from the retrieved documents and then filter based on the presence of the desired answer type within that candidate text. Ranking is then done based on syntactic features such as word order or location and similarity to query.

However, in the cases where simple question reformulation or keyword techniques will not suffice, more sophisticated syntactic, semantic and contextual processing must be performed to extract or construct the answer. These techniques might include named-entity recognition, relation detection, word sense disambiguation, logical inferences (abduction) and commonsense reasoning, temporal or spatial reasoning and so on. These systems will also very often utilize word knowledge that can be found in ontologies such as WordNet, or the Suggested Upper Merged Ontology (SUMO) to augment the available reasoning resources through semantic connections and definitions. These are important and possible directions in the future research.

References

Agirre, E., Rigau, G., 1996. Word Sense Disambiguation Using Conceptual Density. Proceedings of the 16th International Conference on Computational Linguistics (Coling'96). Copenhagen, Denmark, p.16-22.

Banerjee, S., Pedersen, T., 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation Using Wordnet. Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics. Mexico City, p.136-145.

Bruce, B., Wiebe, J., 1994. A New Approach to Sense Identication. ARPA Workshop on Human Language Technology. Plainsboro, NJ.

Chua, S., Kulathuramaiyer, N., 2004. Semantic Feature Selection Using WordNet. Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004), p.166-172. [doi:10.1109/WI.2004.10115]

Fellbaum, C., 1998. An Electronic Lexical Database. MIT Press.

Hirst, G., St-Onge, D., 1998. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. *In*: Fellbaum, C. (Ed.), WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA.

Jiang, J.J., Conrath, D.W., 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. Proceedings of ROCLING X (1997) International Conference on Research in Computational Linguistics. Taiwan.

Kilgarriff, A., 1997. I don't believe in word senses. *Computers and the Humanities*, **31**(2):91-113. [doi:10.1023/A:1000583911091]

Kilgarriff, A., Rosenzweig, J., 2000. Framework and results for English SENSEVAL. *Computers and the Humanities: Special Issue on SENSEVAL*, **34**(1/2):15-48. [doi:10.1023/A:1002693207386]

Kim, S.B., Seo, H.C., Rim, H.C., 2004. Information Retrieval Using Word Senses: Root Sense Tagging Approach. Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval (SIGIR'04). Sheffield, the United Kingdom, p.258-265. [doi:10.1145/1008992.1009038]

Leacock, C., Martin, C., 1998. Combining Local Context with Wordnet Similarity for Word Sense Identification. *In*: Fellbaum, C. (Ed.), WordNet: A Lexical Reference System and Its Application. MIT Press, Cambridge, MA.

Lee, J.H., Kim, M.H., Lee, Y.I., 1993. Information retrieval based on conceptual distance in IS-A hierarchies. *Journal of Documentation*, **49**(2):188-207.

Lesk, M., 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Code from an Ice Cream Cone. Proceedings of the 5th Annual International Conference on Systems Documentation. ACM Press, p.24-26.

Li, H., Li, C., 2004. Word translation disambiguation using bilingual bootstrapping. *Computational Linguistics*, **30**(1):1-22. [doi:10.1162/089120104773633367]

Lin, D., 1998. An Information-theoretic Definition of Similarity. Proceedings of the International Conference on Machine Learning.

Lin, D., 1999. A Case-base Algorithm for Word Sense Disambiguation. Proceedings of Conference Pacific Association for Computational Linguistics. Pacific Association for Computational Linguistics, Waterloo, Canada.

Lin, D., 2000. Word sense disambiguation with a similarity based smoothed library. *Computers and the Humanities: Special Issue on SENSEVAL*, **34**(1/2):147-152. [doi:10.1023/A:1002633105432]

Miller, G.A., 1995. WordNet: a lexical database. *Comm. ACM*, **38**(11):39-41. [doi:10.1145/219717.219748]

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K., 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, **3**(4):235-312.

Moldovan, D., Mihalcea, R., 2000. Using WordNet and lexical operators to improve Internet searches. *IEEE Internet Computing*, **4**(1):34-43. [doi:10.1109/4236.815847]

Patwardhan, S., Banerjee, S., Pedersen, T., 2003. Using Measures of Semantic Relatedness for Word Sense Disambiguation. Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics. Mexico City, p.241-257.

Rada, R., Mili, H., Bicknell, E., Bletner, M., 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, **19**(1):17-30. [doi:10.1109/21.24528]

Resnik, P., 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. Proceedings of the 14th International Joint Conference on Artificial Intelligence. Montreal, **1**:448-453.

Rosso, P., Masulli, F., Buscaldi, D., 2003. Word Sense Disambiguation Combining Conceptual Distance, Frequency and Gloss. Proceedings of International Conference on Natural Language Processing and Knowledge Engineering, p.120-125. [doi:10.1109/NLPKE.2003.1275880]

Slator, B.M., Wilks, Y.A., 1987. Towards Semantic Structures from Dictionary Entries. Proceedings of the Second Annual Rocky Mountain Conference on Artificial Intelligence (RMCAI-87). Boulder, CO, p.85-96.

Sussna, M., 1993. Word Sense Disambiguation for Free Text Indexing Using a Massive Semantic Network. Proceedings of the Second International Conference on Information and Knowledge Management. Arlington, Virginia. [doi:10.1145/170088.170106]

Yang, C.Y., Hung, J.C., Wang, C.S., Chiu, M.S., Yee, G., 2005. Applying Word Sense Disambiguation to Question Answering System for E-Learning. The 19th International Conference on Advanced Information Networking and Applications (AINA 2005). IEEE Computer Society, Taipei, ISBN 0-7695-2249-1.

Yarowsky, D., 1992. Word-sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large-Corpora. Proceedings of the 15th International Conference on Computational Linguistics (Coling'92). Nantes, France.

Yu, J.S., Wen, Z.S., Liu, Y., Jin, Z.H., 2004. Statistical Overview of WordNet from 1.6 to 2.0. The Second Global Wordnet Conference (GWC 2004). Brno, Czech Republic.