In this paper, word sense disambiguation methods utilizing a knowledge base based on a semantic network are introduced. The basic idea is to keep track of a set of paths in the knowledge base which correspond to the incremental semantic interpretation of an input sentence. These paths are called the semantic paths. When the parser reads a word, the senses of this word which are not involved in any of the semantic paths are removed. Then the removal operation is propagated through the knowledge base to invoke the removal of the senses of other words that have been read before. This removal operation is called recursively as long as senses can be removed. This is called the recursive word sense removal. Concretion of a vague word's concept is one of the important word sense disambiguation methods. We introduce a method called the path adjustment that extends the concretion operation. How to use semantic association or syntactic processing in cooperation with the above methods is also considered.

1. Introduction

It is recognized that word sense disambiguation is one of the several natu-
eral language understanding problems that are hard to solve completely. However, humans seem to have no trouble in selecting the correct sense of words in sentences. Psycholinguists have tried to find the mechanism that humans use to achieve this high performance. But this mechanism has not been completely known yet.

There are three kinds of ambiguity related to words: homonymy, polysemy and categorial ambiguity. Homonymous words (for example, ball/toy-ball, ball/dance-party) have multiple meanings (senses) that are not related. Polysemous words (for example, “go”) have several meanings that are related. The senses of a polysemous word have a meaning in common. A polysemous word is also called a vague word. A word is categorically ambiguous when it can be used as more than one syntactic category (for example, rock/shake, rock/stone).

There were two recent research efforts that use knowledge bases built with frame-based knowledge representation. Hirst’s [1983] research is largely related to the disambiguation of homonymous words. He used what he called Polaroid Words (PW) to realize his idea about the gradual disambiguation of ambiguous words. A PW is created and assigned to each word as it is inputed to the parser. PWs can be considered to be communicating processes that cooperate together. A PW first starts with all possible senses of its word. It gradually eliminates the senses that are inappropriate to the surrounding context by communicating with other PWs. When a PW eliminates some of its senses, it announces this. Then, other PWs use this information to try to eliminate some of their senses. This elimination and announcement process goes on until all words have been read and the exchange of information becomes quiescent.

Lytinen [1984, 1988] developed a method to disambiguate the sense of vague words. He used the concept refinement process for this purpose. A vague word is initially represented by a concept that is high in the
hierarchically organized knowledge base. As more words are read in and more contextual information is available, a set of concept refinement rules are used to refine the concept to a more specific concept in the knowledge base. But it is not clear how the homonymous words (especially nouns) are disambiguated using Lytinen’s strategy.

In this paper, we will explain a word sense disambiguation method used in the natural language parser called SYNSEM. Influenced by the work of Charniak[1983, 1986], SYNSEM has been built based on the marker passing mechanism. SYNSEM keeps track of a set of paths in the semantic network. These paths corresponds to semantic interpretation of the input sentence (we call these paths the semantic paths). If a word has multiple senses and it has not been disambiguated yet, there can be false semantic paths being kept due to the inappropriate senses of the ambiguous words. If a word has a vague word sense, then it will be represented by an abstract concept until it is disambiguated to a more specific concept. As parsing goes on, these false semantic paths will be eliminated and abstract concepts will be replaced by more concrete concepts. All the disambiguation strategies adopted by SYNSEM is made possible by using a knowledge based on a semantic network.

2. Overview of the Parser SYNSEM

SYNSEM consists of the three major components: the marker passer (MP), the syntactic analysis component (SYN), and the semantic analysis component (SEM). The knowledge base (KB) that is used by SYNSEM is written in the KODIAK knowledge representation language which is based on the semantic network formalism (for KODIAK, refer to [Norvig, 1986; Wilensky, 1987]).
For each open-class word that is inputted into the parser, the MP passes markers starting from the instance node through the concept nodes that correspond to the senses of the word (the concept nodes are listed in dictionary entry of the word). For example, man.1, ate.2 and apple.3 in Figure 1 are the instance nodes for the words in the sentence (1).

(1) The man ate an apple.

![Figure 1. Representation of words in the KB](image)

Concepts are represented by rectangles in the figure. In Kodiak, concepts are called the absolutes. Ellipses in the figure represent the instances of the concepts of the words mentioned in the input sentence. A relation node shown in double circle is used to represent a semantic relation between two concepts. Specifying semantic relations by explicit nodes called relations is one of the distinct characteristics of Kodiak. The circles in the figure represent the aspectuals. Each relation node is connected to two aspectual nodes that point to the absolutes that are involved in the semantic relation. Thus, two aspectuals connected to a relation node can be considered to be the formal parameters of the semantic relation.

The D-links (Dominate) in the figure specifies the super-class and sub-class relationship. D-links are usually called Is-A links in other knowledge
representation languages. For each open-class word, one instance node is created in the KB to denote the concept of the word. I-links coming out of a instance node points to the absolute nodes that might be the concept of the word. A c-link from an aspectual to an absolute is used to say the fact that the absolute fills the role of the aspectual.

When SYNSEM reads the word man in (1), it first creates the instance node man.1 which is linked to the concept man via an I-link. Then the MP passes markers starting from man.1 through the links in the KB. Note that markers are not passed in the reverse direction of D-links or I-links to prevent markers from propagating too widely. Markers can not flow through D-links between relation nodes.

During marker passing, a collision is reported when a marker is passed to a node that already has a marker which has a different origin. For example, in Figure 1, a collision will be reported on the node eat. This collision is caused by a marker that originated from man.1 and a marker from ate.2. A collision of these two markers will also be reported on the node eater-eat when the marker whose origin is ate.2 reaches eater-eat. But this collision is a redundant collision of the former collision at eat because they correspond to the same paths between the two origins: this redundant collision is called a follow-on collision[Hendler, 1986]. From a collision, one or more paths are formed out of the traces of the two markers involved in the collision.

One of the functions of SYN is to send suggestions of the form(N1 N2 R) to SEM while syntactic analysis goes on. N1 and N2 are instance nodes and R is a preposition indicating the grammatical relationship between the input words corresponding to N1 and N2. R can be a pseudo-preposition("subj" or "obj") or a preposition. For example, SYN will send the suggestion (man.1 ate.2 subj) to SEM during the parse of the sentence (1). But during the parse of the sentence “the man in the room…”’, SYN will send the suggestion (man.1 room.2 in). SYN may send a suggestion that does not contain a preposition
in some cases. For example, when it is analyzing the nominal compound "a screw head", the suggestion (screw.1 head.2) will be sent to SEM because there is no preposition available between the two words, screw and head.

For each suggestion, SEM should find only good paths [Charniak, 1986] between the two nodes in the suggestion. The paths found are registered in the system and they are called semantic paths. For example, the two paths (P1) and (P2) are the semantic paths found after (1) is processed (some of the nodes are not shown for simplicity in these paths).

(P1) (man.1 man person eater–eat eat ate.2)
(P2) (apple.3 fruit food eatee–eat eat ate.2)

(P1) indicates that the entity man.1 is the agent (indicated by the relation eater–eat) of ate.2 which is a specific instance of action eat. (P2) says that the entity apple.3 is a kind of fruit which is again a kind of food and it is the patient (indicated by the eatee–eat relation) of the specific action instance ate. 2.

A preposition provided in a suggestion is useful for selecting good paths. The pseudo–preposition subj in (man.1 ate.2 subj) indicates that man.1 and ate.2 should be connected by paths that pass through the relation node eater–eat because eater–eat is the relation that is implied by the pseudo–preposition subj.

The preposition in contained in (man.1 room.2 in) makes SEM select only the paths that go through the relation node location–in. Without in, it is hard to decide which paths to select out of many paths that can connect man.1 and room.2. To facilitate this path–finding process, a dummy node is created in the KB for each preposition. Then all relation nodes that can be implied by the preposition are made to be reachable from this dummy node via D–links. In Figure 1, eater–eat can be reached from subj. Recipient–receive can also be reached from the node subj.
The KB constructed in this way is utilized by SEM. For the suggestion (man.1 ate.2 subj), SEM passes another kind of marker starting from subj. This is called secondary marker passing in contrast to marker passing done by the MP (called primary marker passing). The secondary markers can flow only through the reverse direction of D-links between relation nodes. Then SEM only consider the collisions of the three markers (two primary and one secondary) from man.1, ate.2 and subj. This is called a three-way collision. Information that prepositions can provide is used in this way in SYNSEM. For a suggestion missing a preposition, SEM does not use secondary marker passing.

3. Word Sense Disambiguation Methods

In this section, several methods for word sense disambiguation used in SYNSEM will be explained. A part of the KB used by the parser is shown in Figure 2. Let us use the sentence (2) [Hirst, 1983] to show how disambiguation is done:

(2) The crook operated the pizza parlor.

Figure 2. A part of the knowledge base for the example
3.1 The basic method for word sense disambiguation

Use of semantic paths for word sense disambiguation will be described in this section. To eliminate the inappropriate senses of a word in a sentence, we utilize the semantic paths that are found during the semantic interpretation of the sentence. A sense can be eliminated if it is not used in any of the semantic paths because it is not used by the semantic interpretation of the sentence. As the semantic interpretation proceeds, some of the semantic paths may be found to be false later and they are removed. Then the senses of the semantic paths which are removed later should be eliminated if they are not used in any remaining semantic paths.

The elimination of a sense of a word is propagated through the semantic paths, which results in the removal of other word senses. Using this idea, we are able to characterize the word sense elimination process more clearly and implement it more efficiently. How our word sense disambiguation method works will be explained in detail using the parse of the input sentence (2).

When the word crook is input, SYNSEM will create the instance node crook. 1. This instance node has I-links pointing to all senses of the word crook as shown in Figure 3.

![Figure 3. Representation of multiple word senses](image)

This means that SYNSEM supports the all-readings hypothesis for the word sense access [Swinney, 1979]. To each I-link, a status flag is attached to indicate the status of the link (the status of an I-link is written in the pa-
rentheses in Figure 3). A status flag can be one of the following values: “i”, “u”, or “r”. “i” means that the link is in the initial state. In other words, it has not been processed yet. “u” indicates that the I-link is currently being used in a semantic path and the corresponding sense is still a candidate for the correct sense of the word. An I-link for the sense that has been eliminated is tagged with “r” (removed). When a word has been disambiguated fully, there should be only one I-link with the status flag “u” emanating from the instance node of a word. This is illustrated in the right side of Figure 3.

For the input word crook, MP passes markers starting from its instance node crook.1. The next word operated is inputted which is represented by the instance node operated.2. Marker passing is done for this word, too. After syntactic analysis, SYN will send a suggestion (crook.1 operated.2 subj) to SEM. SEM will find all possible good paths in the KB as shown in Figure 4. There should be only one good path for a suggestion. But if a word related to the suggestion has multiple senses as in this example, multiple semantic paths can be found.

Three paths have been found in this example as in Figure 4. These paths are stored in the set of semantic paths. Note that shepherd→staff has no path to perform→surgery because the KB in Figure 2 defines that the instrument of performing surgery should be some kind of medical tool and the shepherd's staff is not one of them. At this point of parsing, the two senses of crook are
still used in the semantic paths. Thus, it has not been fully disambiguated yet.

The word *operated* has not been disambiguated either. If we assume that the word “operated” has another sense S1 (the dotted box in the figure) and it is not used in any path found, then the sense S1 will be removed by attaching the flag “r” to the corresponding I-link. For a suggestion, usually one path is found. But three paths have been found in this case because of the ambiguity. As parsing goes on, the two false paths will be removed later.

Now the next input is the canned phrase *pizza parlor* which is represented by the single concept pizza_parlor (let us assume that the syntactic analyzer can detect the canned phrase). SYN will send the suggestion (operated.2 pizza_parlor.3 obj) to SEM. For this suggestion, there is only one path (d) found as shown in Figure 5.

![Figure 5. The effect of reading next words 'pizza parlor'](image)

Note that the sense perform_surgery is not used in this path. This means that perform_surgery cannot be the sense of the word *operated*. The sense perform_surgery is removed by attaching the flag “r” to the corresponding I-link. The removal of the sense perform_surgery makes SEM remove the path (c) in the figure from the set of semantic paths which results in the set of semantic paths shown in Figure 6.

At this point of parsing, the word *operated* has only one sense with “u” flag, which means that it has been fully disambiguated. But there are still two senses with “u” flag for the word crook. Thus, it still has still word sense ambi-
guity.

Figure 6. Paths after removing a sense and a path

If there were not the path (b) between crook.1 and operated.2, then the removal of the path (c) would result in the removal of the sense criminal of the word crook because criminal would not be used in any semantic path connecting crook.1 and operated.2. As soon as a sense gets an "r" flag, it will never be considered by SEM during the path-finding process.

Let us consider a general case which can be illustrated by Figure 7. W.8, Y.1 and X.4 are instance nodes for the words already processed by the parser. W.8 has two senses which are P and Q and these two senses are still candidate senses. A and B are candidate senses for Y.1. D and E are candidate senses for X.4. At this point, the new word Z.6 has just been inputed. The suggestion between Z.6 and X.4 resulted in only one path (e) as shown in Figure 7. The sense D of X.4 is not used in this path (e). Thus, the sense D can not be the sense of X.4 and it is removed.

Figure 7. An example for recursive word sense removal
The removal of the sense D causes the path (d) to be removed from the set of semantic paths. The removal of (d) causes the sense B to be removed from the possible senses of Y.1. The removal of the sense B again causes the removal of the path (a) from the set of semantic paths. This again causes the removal of the sense P from the possible senses of W.8. P is not connected to any other paths and thus the removal operation stops here.

From this process, we can notice that the removal of a word sense propagates recursively through the KB via the semantic paths. We call this operation the recursive word sense removal (RWSR).

But there is a case which requires more careful consideration. Consider the case shown in Figure 8 which has one more path (f) added to Figure 7. During the RWSR operation invoked by the removal of D of X.4, P will be considered for removal after the removal of the path (a). However note that P is also used in the path (f). Thus, P should not be removed. The sense P is an example of the so-called non-removable senses. After the path-finding process for a suggestion has been done, SEM computes the set of non-removable senses before the word sense disambiguation process starts. SEM computes the set of non-removable senses as follows:

Starting from the word senses of the new word that are included in any semantic path found for a suggestion, find all word senses that can be reached via the semantic paths.

In Figure 8, this set will be (F E A Q P). During the RWSR operation, any word sense that is in the set of non-removable senses is not removed.

Figure 8. Consideration for the set of Non-removable senses
In this section, we have been considering the word sense disambiguation process that is initiated by each suggestion sent by SYN. This operation starts from the set of semantic paths that is found by SEM for each suggestion from SYN. Let's assume that $S$ is the set of semantic paths found for a suggestion having old-instance and new-instance as two origins. The word sense disambiguation process is shown in Figure 9:

1. Remove the senses of new-instance that are not in any of the paths in $S$ (for example, the node C in Figure 10).
2. Compute the set Non-removable-senses. (In Figure 10, all senses that are reachable from A or B via semantic paths are included in Non-removable senses.)
3. For each of the senses of old-instance having "u" flag, if that sense is not used in any path in $S$, call RWSR procedure (in Figure 11) passing this sense and old-instance as the argument.

Figure 9. The word sense disambiguation process

Figure 10. Starting the removal operation
Procedure RWSR(Sense, Instance)
    begin
    if Sense is in the set Non-removable-senses
    then exit
    else
    begin
    Remove Sense from the possible senses of Instance;
    Find all sense-instance pairs that can be reached from the pair Sense-Instance
    (given as the input parameters) via any path in S, and insert these pairs
    into the set Reachable, and remove from S the paths used for this finding;
    For each element in Reachable, call RWSR procedure passing the
    element (which is a sense-instance pair) as the argument.
    end;
end;

Figure 11. The RWSR algorithm

3.2 Use of hierarchical information in the KB

In this section, the hierarchical information contained in the semantic network is used as a source of word sense disambiguation. This is usually called the concretion operation. Because of the specialization of some concepts after concretion, some semantic paths should be adjusted, which leads to the path adjustment operation that should be also considered along with concretion.

3.2.1 Concretion operation

Another mechanism used by SYNSEM for word sense disambiguation is the concretion operation. A sense of a word is changed to a more specialized sense when a path of a special shape connecting the two instances in the suggestion are detected. Let's call this special shape the concretion shape. The concretion shape is shown in Figure 12. In Figure 12, the path, (instance2 A B D E F G H instance1), is of the concretion shape. Note that there can be zero or more absolutes between G and instance1 and between A and B in the figure
as indicated with *. According to this concretion path, the sense of instance2 can be specialized from A to B.

![Diagram](image)

**Figure 12. Shape of a concretion path**

The reason for this is as follows: the conceptual association between instance1 and instance2 is via B and A; but B is a specialization of A; thus, instance2 can actually imply the meaning B (which is a specialization of A). The concretion path in Figure 12 is the best path found for the suggestion (instance1 instance2 prep).

The detection of the concretion path is done in a special way. Note that markers from instance1 and instance2 can not create a two-way collision at the relation node E because a marker from instance2 can not flow in the reverse direction of D-links and thus can not reach the node E. But it is necessary to get a two-way collision at E so that a three-way collision can occur at E during secondary marker passing and the path can be found as the best path for the suggestion.

This problem is handled in the following way. Note that the marker starting from instance1 will reach node A via the course of the concretion path (i.e., via H, G, F, E, D, and B). The two-way collision occurs at the node A
when a marker from instance2 reaches the node A. For every two-way collision, the MP checks if the trace of one marker for the collision is of the concretion shape and the trace for another marker of the collision is of the form (collision–node I–link instance–node). Then a dummy two-way collision is put on the relation node which is in the middle of the trace (the node E in Figure 12).

We allow the dummy collision node to be used to form a three-way collision when it collides with a secondary marker. This is the way that the concretion path is found for a suggestion. When a concretion path is found for a suggestion, the sense of an instance is specialized. In the current example, the sense of instance2 is changed from A to B.

Let’s look at the ongoing example that is using (2). In the previous section, it was explained that the path (d) in Figure 6 was found for the suggestion (operated.2 pizza-parlor.3 obj). Actually this path is a path of the concretion shape as shown in Figure 13.

![Figure 13. Concretion for the example](image)

Noticing that (d) in Figure 6 is a concretion path, the sense of operated is changed from cause_function to cause_function_pizza_parlor. This is called the concretion operation.
3.2.2 The path adjustment operation

The concretion operation requires the system to adjust the semantic paths that goes through the old word sense (cause-function in this example). For example, the path (a) and (b) in Figure 6 goes through cause-function to get to operated.2 from crook.1. Thus, these paths should be adjusted so that they can go through the concreted sense, cause-function_pizza_parlor, instead of the old sense cause-function. This is called the path adjustment operation invoked by the concretion operation.

It is important to recognize that the path adjustment operation after concretion is also a source of word sense disambiguation information. To see this, let us consider how the path adjustment operation is done in the ongoing example.

The new adjusted path should satisfy the following condition: the relation node of the new path is the specialization of (or the same as) the relation node of the old path and the end absolutes of the old path is the ancestor of (or the same as) the end absolutes of the new path (end absolutes are absolutes that come first or last in a path). If a new path satisfying this condition can be found, the new path is added to the set of semantic paths replacing the old path. If the new adjusted path can not be found, the old path is simply thrown away.

![Figure 14. Success of path adjustment](image-url)
The adjustment of the path (b) in Figure 6 is shown in Figure 14. The new adjusted path shown in this figure satisfies the condition stated above. In the case of the adjustment operation for the path (a) in Figure 6, the new adjusted path satisfying the above condition cannot be found (See Figure 15). The reason is twofold: shepherd_staff is not an ancestor of capital (meaning money) and there is no connection between capital and crook.1. Therefore, the path (a) in Figure 6 is removed from the set of semantic paths, which results in the removal of shepherd_staff from the sense of crook.1. At this point, the word crook has only one sense criminal and is fully disambiguated.

![Diagram](image)

*Figure 15. Failure of path adjustment*

The path adjustment operation after concretion is a new source of disambiguation information first introduced in SYNSEM. The disambiguation by the path adjustment operation can also invoke the RWSR operation to remove other senses, which means that the different sources of disambiguation cooperate each other.

3.3 Use of semantic association

Semantic association is one of the sources of information that is useful for word sense disambiguation. It seems that Quillian[1968] is the first
researcher who used it. Semantic association can be modeled naturally in a semantic network. Two concepts have a semantic association if there is a path connecting them in the network. A concept is said to be semantically primed by another concept if the access of the latter concept facilitates the process (or access) of the former concept. In a marker passing system, nodes X and Y have semantic association if markers originating from them collide at a node. If X gets a marker whose origin is Y, it can be said that X is semantically primed by Y.

SYNSEM uses these notions for disambiguation. When the MP begins to do marker passing for a new input word having multiple senses, it checks if there is a sense of this word that has been primed by a previous word. Then the MP removes the other senses of the new word by tagging the corresponding I-links with “prime-removed” (this is an additional possible value of the status flag). The MP does not pass markers from the senses with a “prime-removed” tag, which implies that these senses are not accessed. This is to follow the research in [Seidenberg etc., 1982] which found that semantic priming allows only the primed sense to get accessed.

An ambiguous word which still has more than one sense remaining can be disambiguated by a new input word, if there is a strong semantic association (detected by the MP) between one sense of the ambiguous word and one sense of the new word. “Strong” semantic association is defined as follows in SYNSEM: there should be a path that does not involve more than two relation nodes; D-links are allowed only at both sides of the path but they should be only for going downward in the KB to get to the instance node; and a very general concept (with high promiscuity) cannot be used in the path.

The MP removes the senses of the ambiguous word that are not involved in the strong semantic association by tagging them with “prime-removed”. This process can be explained using the input sentence [Hirst, 1983]:

(3) The slug operated the vending machine.
We assume that the possible senses of \textit{slug} are “a gastropod without shell”, “a fake coin”, “a bullet” or “a shot of liquor”. After \textit{operated} is processed, the following three semantic paths are found and registered:

(4) (slug.1 fake-coin coin inanimate instr-c.function cause-function operated.2)

(5) (slug.1 bullet inanimate instr-c.function cause-function operated.2)

(6) (slug.1 shot-liquor inanimate instr-c.function cause-function operated.2)

Note that perform-surgery is removed from the possible candidates for the sense of \textit{operated} because it is not used in any of the paths. The sense gastropod-without-shell is removed, but three senses (fake-coin, bullet, and shot-liquor) are still the possible candidates for the correct sense of the word \textit{slug}. After the \textit{vending machine} is input, the MP finds the following path indicating strong semantic association between slug.1 and vending-machine.3 through the sense fake-coin:

(7) (vending-machine.3 vending-machine obj-c.function-vending-machine cause-function-vending-machine instr-c.function-vending-machine coin fake-coin slug.1)

Thus fake-coin is chosen as the sense of \textit{slug}. Other senses and their corresponding paths, (5) and (6), are removed. This process is shown in Figure 16. Note that the removal of the sense A in the figure also invokes the RWSR operation explained in the previous section.

3.4 Use of syntactic information

To each I-link of an instance node is attached the “part of speech” flag whose value can be one of noun, verb, adjective, etc. When SYN determines the part of speech of a word, all the I-links of the instance node for this word that do not correspond to the correct part of speech are removed by tagging
them with "r" (removed). This removal of some senses can also lead to the recursive word sense removal operations. The word senses that have been removed will never be used by SEM. How to determine the correct part of speech can not be explained because of space limitation.

![Diagram](image)

**Figure 16. Use of semantic association**

4. Conclusion

In this paper, we have shown that a semantic network-based knowledge base can be used to facilitate word sense disambiguation in natural language parsing. A new method for applying the word sense elimination mechanism has been proposed. The basic strategy is to keep track of a set of paths in the knowledge base (called semantic paths) corresponding to semantic interpretation of the input sentence. Senses of a word that are not used in the semantic paths for a suggestion from the syntactic analysis component are removed. The removal of a sense at one end of a semantic path results in the removal of all the senses that can be reached via the semantic paths. This is called the recursive word sense removal operation.

A concretion mechanism similar to the concept refinement process is also incorporated into the disambiguation method of the our parser. We have developed a new source of information for word sense disambiguation called the
path adjustment operation to extend the concretion mechanism. The use of marker passing enables the parser to use semantic association and semantic priming as another source of disambiguation information. The information from syntactic analysis also leads to the disambiguation of some word senses which can also invoke the recursive word sense removal operation.

References


의미넷워크를 이용한 단어의미의 모호성 해결방법

나동렬
연세대학교 전산학과

본 논문에서는 의미넷워크에 기반을 둔 지식베이스를 이용하여 단어의미의 모호성을 해소하는 방법들을 소개한다. 기존이 되는 방법은 입력문장의 의미해석이 진행됨에 따라 수집되는 지식베이스내의 경로(path)들을 추적하여 이용하는 것이다. 이러한 경로들을 의미 경로(semantic path)라고 부른다. 파생과정에서 한 단어가 입력되면 이 단어가 가질 수 있는 의미 중에서 어느 의미 경로에도 이용되지 않은 것이 제거된다. 각 제거는 의미 경로들을 통하여 전파되어 이미 입력된 다른 단어들의 의미의 제거를 유발한다. 이 작업은 더 이상 제거되는 의미가 없을 때까지 반복 진행되는데 이를 recursive word sense removal 작업이라 부른다. 추상적인 개념의 구체화(concretion) 작업도 단어의미 모호성 해소의 중요한 방법인데, 본 논문에서는 이를 경로조절작업(path adjustment operation)이라고 불리는 방법을 이용하여 확장시키는 방법을 소개한다. 의미 사이의 연관성(semantic association)이나 구분분석으로부터의 정보를 위의 방법들과 관련지어 이용하는 방법도 살펴본다.

Dongyul Ra
Department of Computer Science
Yonsei University
Maeji, Hung-up, Wonju-Kun
KangWon, 222-701, Korea