Automatic Learning for Semantic Collocation

Satoshi SEKINE*
Tokyo Information and Communications Research Laboratory
Matsushita Electric Industrial Co.,Ltd.
3-10-1, higashimita, tama-ku, kawasaki 214 JAPAN

Jeremy J.CARROLL
Sofia ANANIADOU
Jun'ichi TSUJII
Centre for Computational Linguistics
University of Manchester Institute of Science and Technology
P.O.Box 88, Manchester M60 1QD, United Kingdom

Abstract

The real difficulty in development of practical NLP systems comes from the fact that we do not have effective means for gathering "knowledge". In this paper, we propose an algorithm which acquires automatically knowledge of semantic collocations among "words" from sample corpora.

The algorithm proposed in this paper tries to discover semantic collocations which will be useful for disambiguating structurally ambiguous sentences, by a statistical approach. The algorithm requires a corpus and minimum linguistic knowledge (parts-of-speech of words, simple inflection rules, and a small number of general syntactic rules).

We conducted two experiments of applying the algorithm to different corpora to extract different types of semantic collocations. Though there are some unsolved problems, the results showed the effectiveness of the proposed algorithm.

1 Introduction

Quite a few grammatical formalisms have been proposed by computational linguists, which are claimed to be "good" (declarative, highly modular, etc.) for practical application systems in NLP. It has also been claimed that extra-linguistic, domain specific knowledge is indispensable in most NLP applications, and computational frameworks for representing and using such domain knowledge have also been developed.

However, the real difficulty in developing practical NLP systems is due to the fact that we do not have effective means for gathering the "knowledge", whether linguistic or extra-linguistic. In particular, it has been reported [Ananiadou, 1990] that not only extra-linguistic, domain knowledge but also linguistic knowledge required for application systems varies, depending on text-type (technical reports, scientific papers, manuals, etc.), subject domain, type of application (MT, automatic abstraction, etc.) etc. This means that we have to have effective and efficient methods either for adapting already existing knowledge for a specific "sublanguage" or for acquiring knowledge automatically, for example from sample corpora of given applications.

In this paper, we propose an algorithm which automatically acquires knowledge of semantic collocations among "words". "Semantic" here means that the collocations the algorithm discovers are not collocations among words in the sense of traditional linguistics but collocations that reflect ontological relations among entities in given subject domains. We expect that the knowledge to be extracted will not only be useful for disambiguating sentences but also will contribute to discovering ontological classes in given subject domains.

Though several studies with similar objectives have been reported [Church, 1988], [Zernik and Jacobs, 1990], [Calzolari and Bindi, 1990], [Garside and Leech, 1985], [Hindle and Rooth, 1991], [Brown et al., 1990], they require that sample corpora be correctly analyzed or tagged in advance. It must be a training corpus, which is tagged or parsed by human or it needs correspondence between two language corpora. Because their preparation needs a lot of manual assistance or an unerring tagger or parser, this requirement makes their algorithms troublesome in actual application environments. On the other hand, the algorithm in this paper has no such requirement, it requires only a minimum of linguistic knowledge, including parts-of-speech of words, simple inflection rules, and a small number of general syntactic rules which lexicon based syntactic theories like HPSG CG etc. normally assume. The parser is not a deterministic parser, but a parser which produces all possible analyses. All of the results are used for calculation and

*SEKINE is now a visitor at C.C.L., U.M.I.S.T. *sekine@ccl.umist.ac.uk*

the system assumes that there is a correct answer among them. The algorithm builds correct structural descriptions of sentences and discovers semantic collocations at the same time. It works as a relaxation process.

2 Overview

Before giving the algorithm formally, we illustrate an overview in this section, by using simple examples. Though the algorithm can be used to extract knowledge useful to resolve a wide range of syntactic ambiguities, we use here the prepositional phrase attachment problem as an illustrative example.

We assume a syntactic parser which provides all possible analyses. It produces syntactic descriptions of sentences in the form of syntactic dependency structures. That is, the description to be produced is represented by a set of tuples like `[head word, syntactic relation, argument]`, each of which expresses a dependency relation in the input. The syntactic relation in a tuple is either a grammatical relation like SUBJ, OBJ, etc. (in case of a noun phrase) or a surface preposition like BY, WITH, etc. Following the normal convention of dependency representation, the argument is represented by the governor of the whole phrase which fills the argument position.

When an input sentence has attachment ambiguities, two or more tuples share the same argument and the same syntactic-relation but have different head-words.

For example, the description of the sentence

`"I saw a girl with a scarf."`

contains two tuples like

`[girl, WITH, scarf]`
`[saw, WITH, scarf]`

As repeatedly claimed in natural language understanding literature, in order to resolve this ambiguity, a system may have to be able to infer "`a scarf cannot be used as an instrument to see`", based on extra-linguistic knowledge. A practical problem here is that there is no systematic way of accumulating such extra-linguistic knowledge for given subject fields. Furthermore, the ambiguity in a sentence like "`I saw a girl with a telescope`" cannot be resolved only by referring to knowledge about the world. It requires a full range of context understanding abilities, because the interpretation of "`a girl with a telescope`" is less likely in general but can be a correct one in certain contexts. That is, unless a system has a full range of contextual understanding abilities (which we think will be impossible in most application environments in the foreseeable future), it cannot reject either of the possible interpretations as "impossible". The best a system can do, without full understanding abilities, is to select more plausible ones or reject less plausible ones. This implies that we have to introduce a measure by which we can judge plausibility of "interpretations".

The algorithm we propose computes such measures from a given sample corpus in a certain way. It gives a plausibility value to each possible tuple, based on the sample corpus. For example, the tuples `(saw, WITH, scarf)` and `(girl, WITH, scarf)` might be assigned 0.5 and 0.82 as their plausibility value, which would show `(girl, WITH, scarf)` to be more plausible than `(saw, WITH, scarf)` . This produced knowledge can be used to disambiguate interpretations of the sentence "`I saw a girl with a scarf`".

The algorithm is based on the assumption that the ontological characteristics of the objects and actions denoted by words (or linguistic expressions in general) and the nature of the ontological relations among them are exhibited, though implicitly, in sample texts.

For example, nouns denoting objects which belong to the same ontological classes tend to appear in similar linguistic contexts (for example, in the same argument positions of the same or similar verbs). Or if an object (or an ontological class of objects) is "intrinsically" related to an action (like "`telescope`" to "`see`"), the word denoting the class of objects co-occurs frequently with the verb denoting the action. The co-occurrence would be more frequent than that of those whose ontological relations are rather fortuitous, like "`girl`" and "`telescope`".

Note that we talk about extra-linguistic "ontology" for the sake of explaining the basic idea behind the actual algorithm. However, as you will see, we do not represent such things as ontological entities in the actual algorithm. The algorithm counts frequencies of co-occurrences among words and calculates word distances which interpret such co-occurrences as contexts. Nor do we posit any dichotomy between "intrinsic" relations and "accidental" relations among actions and objects. Differences are quantitative, not qualitative. That is, co-occurrences of "girl" and "scarf" are more frequent than, for example, those of "pig" and "scarf".

The algorithm in this paper computes the plausibility value of hypothesis-tuples like `(girl, WITH, scarf)`, `(saw, WITH, scarf)`, etc., basically by counting frequencies of instance-tuples `[girl, WITH, scarf]`, `[saw, WITH, scarf]`, etc. generated from sample texts by a syntactic parser.

3 Algorithm

3.1 Relaxation Process - Informal Explanation of the Algorithm

Though the algorithm simply counts frequencies of co-occurrences of word relations, there are some complications. In this section, we also use the prepositional phrase attachment problem as an example, though the algorithm can be applied to any kind of structural ambiguity.

1. We have to count frequencies of "meaningful" co-occurrences between verbs and nouns, i.e. co-occurrences where the nouns actually appear in the position of the head-noun of PP's which can be attached to verbs or other nouns. The frequency of "general" co-occurrences where the two words occur, for example, in the same sentences may be of little use.

   This means that we encounter the problem of the chicken and the egg here, i.e. in order to obtain fre-

quencies of "meaningful" co-occurrences in sample texts, we have to know the correct attachment positions of PPs, and determining the correct attachments of PPs in sample texts requires knowledge of frequencies of "meaningful" co-occurrences.

2. We usually cannot expect to have a corpus of sample sentences large enough for "intrinsic" relations to appear significantly more often than "accidental" relations. It is desirable, or inevitable in a sense, to introduce methods of increasing the number of co-occurrences.

   One possible way of doing this (which we have adopted in our algorithm) is to introduce "semantic" similarity measures between words, and count the number of extended co-occurrences taking the similarity measures into account. That is, the frequency of `[girl, WITH, necklace]` in sample texts contributes not only to the plausibility value of the tuple `(girl, WITH, necklace)`, but also to that of `(girl, WITH, scarf)`, according to the similarity value (or semantic distance) of `"scarf"` and `"necklace"`.

   Because we compute semantic distances among nouns based on (dis-)similarities of their patterns of co-occurrence with other words (in short, two nouns are judged to be close to each other, if they often co-occur with the same words), we also encounter an chicken and egg problem here. The calculation of semantic distance requires frequencies of collocation, and in order to find semantic collocations, semantic distance could be helpful.

   The two chicken and egg problems in the above are treated differently in the algorithm. We focus on the first problem in this paper, while readers who are interested in the second problem can refer to [Sekine et al., 1992]

   In the following, we call the tuples generated from sample texts by a parser "instance-tuples" and the tuples to which plausibility value are assigned "hypothesis-tuples". Instance-tuples and hypothesis-tuples are indicated by [A, R, B] and (A, R, B), respectively.

   Note that for the sake of explanation the following is not an accurate description of the algorithm. An accurate one is given in the next section.

```
Input: I saw a girl with a telescope.
```

(STEP-1) Generate instance-tuples

All possible instance-tuples such as `[saw, SUBJ, I]`, `[girl, WITH, telescope]`, `[saw, WITH, telescope]`, etc. are generated by a simple parser.

(STEP-2) Assign credits

Assign credits to the instance-tuples, by considering the plausibility value of corresponding hypothesis-tuples. As we will explain later, we assign credits in such a way that

(a) the sum of credits assigned to competing instance-tuples is equal to 1. Competing tuples means such tuples as `[girl, WITH, scarf]` and `[saw, WITH, scarf]` which show different attachment positions of the same PP.

(b) the credits assigned to instance-tuples are proportional to the plausibility value of the corresponding hypothesis-tuples.

   Because hypothesis-tuples have the same plausibility value at the initial stage, each instance-tuple is assigned the same credit, say, 1/(number of competing tuples). The credit of `[saw, SUBJ, I]` is one, while the credits of `[girl, WITH, scarf]` and `[saw, WITH, scarf]` are 0.5.

(STEP-3) Calculate plausibility values

Compute plausibility values of hypothesis-tuples, accumulating credits assigned to the corresponding instance-tuples.

All occurrences of instance-tuples generated from the sample corpus have their credits assigned in (STEP-2). We assume that tuples corresponding to "intrinsic" ontological relations occur more often in texts than "accidental" ones. That is, we expect that instance-tuples of `[girl, WITH, scarf]` occur more often than those of `[saw, WITH, scarf]` and that the sum of the credits of `[girl, WITH, scarf]` is greater than that of `[saw, WITH, scarf]`. This leads to a higher plausibility value for `(girl, WITH, scarf)` than for `(saw, WITH, scarf)`.

After (STEP-3), the algorithm goes back to (STEP-2) to compute new credits to instance-tuples. Unlike the first cycle, because the hypothesis-tuple `(girl, WITH, scarf)` has been assigned a higher plausibility value than `(saw, WITH, scarf)`, the credit to be assigned to `[girl, WITH, scarf]` would be higher than `[saw, WITH, scarf]`.

When we recompute the plausibility value in (STEP-3), the increased credit assigned to `[girl, WITH, scarf]` in (STEP-2) increases the plausibility value of `(girl, WITH, scarf)` and on the other hand, the decreased credit of `[saw, WITH, scarf]` results in a lower plausibility value for `(saw, WITH scarf)`.

By repeating (STEP-2) and (STEP-3), we expect there should be an increase in the credits assigned to instance-tuples which correspond to correct attachment position. Further, the credits of hypothesis tuples should approach values which represent the real "intrinsicality" of the denoted relationships.

(STEP-3) will be further augmented by introducing semantic distances between words., i.e. a similar hypothesis helps to increase the credit of a hypothesis. We expect this should resolve the second type of chicken and egg problems. See [Sekine et al 1992]

3.2 Terminology and notation

instance-tuple $[h, r, a]$: a token of a dependency relation; part of the analysis of a sentence in a corpus.

hypothesis-tuple (h, r, a) : a dependency relation; an abstraction or type over identical instance-tuples.

cycle : repeat time of the relaxation cycle.

$C_{T,i}$: Credit of instance-tuple T with identification number i. $[0, 1]$

V_T^g : Plausibility value of a hypothesis-tuple T in cycle g. $[0, 1]$

$D^g(w_a, w_b)$: distance between words, w_a and w_b in cycle g. $[0, 1]$

3.3 Algorithm

1. For each sentence we use a simple grammar to find all tuples possibly used in this sentence. Each instance-tuple is then given credit in proportion to the number of competing tuples.

$$C_T = \frac{1}{number\ of\ competing\ tuples} \quad (1)$$

This credit shows which rules are suitable for this sentence. On the first iteration the split of the credit between ambiguous analyses is uniform as shown above, but on subsequent iterations plausibility values of the hypothesis-tuples V_T^{g-1} before the iteration are used to give preference to credit for some analyses over others. The formula for this will be shown later.

2. Hypothesis-tuples have a plausibility value which indicates their reliability by a figure from 0 to 1. If an instance-tuple occurs frequently in the corpus or if it occurs where there are no alternative tuples, the plausibility value for the corresponding hypothesis must be large. After analyzing all the sentences of the corpus, we get a set of sentences with weighted instance-tuples. Each instance-tuple invokes a hypothesis-tuple. For each hypothesis-tuple, we define the plausibility value by the following formula. This formula is designed so that the value does not exceed 1.

$$V_T^g = 1 - \prod_i (1 - C_{T,i}) \quad (2)$$

3. At this stage, the word-distances can be used to modify the plausibility values of the hypothesis-tuples. The word-distances are either defined externally by human intuition or calculated in the previous cycle with the formula shown later. Distance between words induces a distance between hypothesis-tuples. Then for each hypothesis-tuple, another hypothesis-tuple which gives greatest effect can be used to increase its plausibility value. The new plausibility value with similar hypothesis-tuple effect is calculated by the following formula.

$$V_T^g = V_T^g + (1 - V_T^g) * V_{T'}^g * (1 - D^g(w_a, w_b))^2 \quad (3)$$

Here, the hypothesis-tuple T' is the hypothesis-tuple which gives the greatest effect to the hypothesis-tuple T (original one). Hypothesis-tuple T and T' have all the same elements except one. The distance between T and T' is the distance between the different elements, w_a and w_b. Ordinarily the difference is in the head or argument element, but when the relation is a preposition, it is possible to consider distance from another preposition.

4. Distances between words are calculated on the basis of similarity between hypothesis-tuples about them. The formula is as follows:

$$D^g(w_a, w_b) = \frac{\sum_T (V_T^g - V_{T'}^g)^\beta}{n} \quad (4)$$

T and T' are hypothesis-tuples whose arguments are w_a and w_b, respectively and whose heads and relations are the same. β is a constant parameter.

5. This procedure will be repeated from the beginning, modifying the credits of instance-tuples between ambiguous analyses by using the plausibility values of hypothesis-tuples. This will hopefully be more accurate than the previous cycle. On the first iteration, we used just a constant figure for the credits of instance-tuples. But this time we can use the plausibility value of the hypothesis-tuple which was deduced from the previous iteration. Hence with each iteration we expect more reliable figures. To calculate the new credit of instance-tuple T, we use:

$$C_T = \frac{V_T^{g\alpha}}{\sum_T (V_T^{g\alpha})} \quad (5)$$

Here, V_T^g in the numerator position is the plausibility value of a hypothesis-tuple which is the same tuple as the instance-tuple T. V_T^g in the denominator position are the plausibility values of competing hypothesis-tuples in the sentence and the plausibility value of the same hypothesis-tuple itself. α is a constant parameter.

6. Iterate step 1 to 5 several times, until the information is saturated.

4 Experiment

We conducted two experiments to show the effectiveness of our algorithm. The first one uses a small, artificial corpus to show how the algorithm works. The second one is a real experiment in which we use data from a real corpus (computer manuals).

4.1 Artificial corpus

We treat the prepositional attachment ambiguity in this experiment. Though the corpus consists of only 7 artificial sentences, this experiment shows the basic characteristics and the effectiveness of the algorithm.

The corpus and the input data to the algorithm are as follows:

```
Sentences:
    I saw a girl with a telescope.
    I saw a girl with a scarf.
    I saw a girl with a necklace.
    I saw the moon with a telescope.
    I meet a girl with a telescope.
    A girl with a scarf saw me.
    I saw a girl without a scarf.

Dictionary:
    I, girl, moon, telescope,
    scarf, necklace, me = Noun
    saw, meet = Verb
    with,without = Preposition

Distances:
    0.2 = {with without}
    0.2 = {scarf necklace}
    0.3 = {saw meet}
    1.0 = between unspecified words

Rules of Grammar:
    Noun <-(SUBJ)- Verb
    Verb -(OBJ)-> Noun
    Noun -(Prep<with>)-> Noun
    Verb -(Prep<with>)-> Noun
```

Table 1 shows the result of the first cycle. Figures in this table show the plausibility values of hypothesis-tuples between the words in the corresponding columns. The plausibility value of the hypothesis-tuple (saw, WITH, telescope), for example, is 0.75.

	telescope	scarf	necklace
saw WITH	0.75	0.50	0.50
girl WITH	0.75	1.00	0.50
moon WITH	0.50	-	-
meet WITH	0.50	-	-
saw WITHOUT	-	0.50	-
girl WITHOUT	-	0.50	-

Table 1: Plausibility values after the first cycle

These plausibility values basically reflect the number of co-occurrences of the two words. However, the hypothesis-tuple (girl, WITH, scarf) has plausibility value 1.0, because in the sentence "A girl with a scarf saw me", there is no ambiguity in the attachment position of "with a scarf".

Then we compute the effects of similar hypothesis-tuples by considering distances among words. The effects which the existence of similar hypotheses has on other hypotheses are clearly shown in Table 2.

The plausibility values of the hypothesis-tuples have changed from the former ones. For example, we can find a sharp distinction between the plausibility values of the hypothesis-tuples (saw, WITH, necklace) and (girl, WITH, necklace) Though these two have the same plausibility value 0.50 before considering the effect of similar hypotheses, the plausibility value of (girl, WITH, necklace) becomes 0.82 while that of (saw, WITH, necklace) becomes only 0.66. The difference in the behavior of these two hypothesis-tuples is caused by the difference of the plausibility values assigned to the hypothesis-tuples (girl, WITH, scarf) and (saw, WITH, scarf). The plausibility values are 1.00 and 0.50 respectively.

	telescope	scarf	necklace
saw WITH	0.81	0.66	0.66
girl WITH	0.75	1.00	0.82
moon WITH	0.50	-	-
meet WITH	0.68	0.24	0.24
saw WITHOUT	0.48	0.66	0.32
girl WITHOUT	0.48	0.82	0.32

Table 2: Plausibility values with similar hypothesis effect

Then we proceed to the second cycle, using the plausibility values which were produced in the previous cycle. Table 3 shows the plausibility values after the fifth cycle.

	telescope	scarf	necklace
saw WITH	1.00	0.26	0.30
girl WITH	0.93	1.00	0.99
moon WITH	0.00	0.00	0.00
meet WITH	0.57	0.04	0.04
saw WITHOUT	0.64	0.01	0.01
girl WITHOUT	0.58	1.00	0.64

Table 3: Plausibility values after the fifth cycle

By the fifth cycle, most of the figures have moved well towards the extremes, either 0 or 1.

For example, the plausibility values of the hypothesis-tuples (saw, WITH, necklace) and (girl, WITH, necklace), are well apart, 0.30 and 0.99, respectively, although they had the same plausibility value after the first cycle. Also the hypothesis-tuple (moon, WITH, telescope) has the plausibility value 0.00, though its initial plausibility value was 0.50. We can claim that the learning process has worked well in making these differences.

On the other hand, if the movement towards extreme values was too strong, there might be a possibility that only the strongest plausibility value survived. When there are two hypotheses which have instances in the same sentence, how do the plausibility values move? This can be seen with the two hypothesis-tuples (saw, WITH, telescope) and (girl, WITH, telescope) which are contradictory hypothesis-tuples in the sentence "I saw a girl with a telescope" In the results, both of their plausibility values are high and avoid the monopoly, because a number of instance tuples and a similar hypothesis contribute to increase the plausibility values of both of the hypotheses tuples.

Note that the relation (saw, WITHOUT, telescope) which does not appear in the sentences, has a rather high plausibility value, 0.64. This occurred because of the effect of the similar hypothesis-tuple (saw, WITH, telescope). But the relation (meet, WITH

telescope), which has a relatively high plausibility value 0.57, is normally unacceptable. This is caused by the close distance between the words 'meet' and 'see'.

The distances between words in the 5th cycle are shown in Table 4.

	telescope	scarf	necklace
telescope	0.00	0.54	0.52
scarf	0.54	0.00	0.26
necklace	0.52	0.26	0.00

Table 4: Distances between words in the 5th cycle

These results, both the plausibility values of hypothesis-tuples and the word distances, seem to behave as we expected.

4.2 The Japanese compound noun corpus

We conducted an experiment using compound nouns extracted from a Japanese computer manual, because of its simplicity and feasibility. The corpus consists of 4152 sentences (about 90,000 words). This might be small considered for statistical analysis purpose, but as the corpus is a sublanguage one, the structures of sentences are rather homogeneous and therefore the number of sentences might be considered sufficient.

There are 616 compound nouns in the corpus, where 210 different words appear. We call an element word of a compound noun a 'word'.

No. of words	No. of compound nouns
2	474
3	113
4	27
5	2
total	616

Table 5: Number of compound nouns

We assume that all words in each compound noun can be structurally related, if they satisfy a condition that a relation has a preceding argument and a following head. For example, from a compound noun with 4 elements, we can extract 6 tuples as follows.

Compound noun: N1 N2 N3 N4

tuples	initial credit
[N2, MODIFY, N1]	1/3
[N3, MODIFY, N1]	1/3
[N4, MODIFY, N1]	1/3
[N3, MODIFY, N2]	1/2
[N4, MODIFY, N2]	1/2
[N4, MODIFY, N3]	1

We know that each element can be the argument in one relation. In the above example, N1 has 3 instance-tuples in which to be the argument. We put the credit 1/3 as initial credit for each instance-tuple. Similarly, we put the credit 1/2 for each instance-tuple in which N2 is an argument.

We have not made any word distance information before the process.

We classified the results obtained as correct or incorrect. 'Correct' means that a hypothesis-tuple which has the highest plausibility value is the correct tuple according to our human judgement. 'Incorrect' means it is judged wrong by a human. 'Indefinite' means that plausibility values of some hypothesis-tuples have the same highest value. 'Uncertain' means that it is impossible even for a human to judge which hypothesis tuple is the best without context. The results are shown in Table 6.

Words	correct	incorrect	indefinite	uncertain
3	66	29	5	13
4	41	7	5	1
5	4	0	0	2
total	111	36	10	16
(%)	(70.7)	(22.9)	(6.4)	(-)

Table 6: Results of experiment with compound nouns

The percentage of correct answers was about 70 %. Though this result is not as impressive as that of the last experiment, it is not bad.

From a perusal of the incorrect analyses, we can find typical reasons for making an incorrect analysis. When there are 2 competing tuples for a 3-element compound noun, these tuples are individually both acceptable in many cases. For example, let's take a compound noun 'file transfer operation'. If we consider the two instance-tuples, [transfer, MODIFY, file] and [operation, MODIFY, file], both are acceptable in the absence of any context. In this case, the plausibility values of the two hypothesis-tuples become almost the same. But there might be a very small difference which may be caused by the effect of a similar hypothesis-tuple. If the wrong hypothesis tuple gains the higher plausibility value, the analysis becomes wrong.

We think that the relation between the words of a compound noun can be defined not only by a semantic relation between each word but also by the structure of the compound noun itself. This feature of compound nouns makes it hard to get a higher percentage of correct answers in this experiment.

5 Unsolved Problems

(a) Parameters

The behavior of the algorithm changes according to the two parameters, α in formula 5 and β in formula 4. Though the parameters are set as $\alpha = 4.0$ and $\beta = 20.0$ in the experiments, we have no established procedures for determining these parameters appropriately. We need to develop criteria or methods to determine these parameters, depending on characteristics of sample texts, etc..

(b) Word sense ambiguity

The entity of collocational relation is represented by a word and the relation labels are either a simple grammatical functions or a surface prepositions. This means we ignored the word sense ambiguity of a word or a preposition in this algorithm. A new method to treat this problem might be needed.

(c) Combination with other clues of disambiguation

It is already known that ontological knowledge is not the only clue to settle ambiguities. There are the problems related with context, discourse, situation etc.. We want to weave these problems into our algorithm. It also has to be noted that rather local, structural preferential clues may help disambiguations. [Wilks, 1985]

(d) Word distance

Though we currently assume that the semantic distances of words are given in the form of single numbers, our research group is now planning to extend to cover multi-dimensional aspects of word meanings. This extension may introduce another complication in our algorithm.

(e) Form of collocation

In the current algorithm, semantic collocations are represented in the form of a triplet (tuple). However, each tuple expresses only a collocation between two words. This is not sufficient for treating relationships among several words, such as subcategorization frames of predicates, knowledge frames, etc. In order to treat such multi-word collocations, we may have to treat co-occurrences of triplets in similar fashions to how we treat co-occurrences of words.

6 Future Directions

Besides planning to resolve the problems written above, there are some other ideas for extending our project. Some of them are really stimulating.

(a) More experiments

Though the results of the two preliminary experiments look promising, we have to conduct more experiments using another real corpora before claiming that the algorithm is effective.

(b) Extension to Machine Translation

Though the algorithm in its present form is designed to acquire monolingual knowledge, we are planning to develop it for acquiring "knowledge" for translation.

If "semantic" collocations discovered by the algorithm reflect the domain ontology, the collocations in two languages (and the semantic classes of words to be produced based on the collocations) are expected to be similar in the sense that their correspondence is rather straightforward.

Experience in MT research, however, generally indicates the opposite, i.e. monolingual regularities and bilingual regularities are sometimes orthogonal and the correspondences of two languages are not so straightforward.

These two rather contradicting predictions (and experiences) have to be consolidated through actual experiments.

(c) Incremental learning system

We don't need to distinguish the knowledge acquisition phase from the phase of using it in actual application systems. It is possible to acquire knowledge and exploit it at the same time.

7 Acknowledgements

We would like to thank our colleagues at CCL, in particular Mr.J.Phillips, Mr.K.Kageura, Mr.S.Kinoshita and Miss.E.van de Veen, whose comments on various occasions have been very useful.

References

[Ananiadou, 1990] Sofia Ananiadou. Sublanguage studies as the Basis for Computer Support for Multilingual Communication. *Proceedings of Termplan '90, Kuala Lumpur*, 1990.

[Church, 1988] Kenneth Ward Church. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 16(1)22-29, March 1990.

[Zernik and Jacobs, 1990] Uri Zernik and Paul Jacobs. Tagging for Learning: Collecting thematic relations from Corpus. *13th COLING-90*, 1990

[Calzolari and Bindi, 1990] Nicoletta Calzolari and Remo Bindi. Acquisition of Lexical Information from a large Italian Corpus. *13th COLING-90*, 1990

[Garside and Leech, 1985] Roger Garside and Fanny Leech. A Probabilistic Parser *2nd Conference of the European Chapter of the A.C.L.*, 1985.

[Hindle and Rooth, 1991] Donald Hindle and Mats Rooth. Structural Ambiguity and Lexical Relations. *29th Conference of the A.C.L.*, 1991.

[Brown et al., 1990] Peter Brown, John Cocke, Stephen A.Della Pietra, Vincent J.Della Pietra, Fredrick Jelinek, John D.Lafferty, Robert L.Mercer, Paul S.Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2) 79-85, 1990

[Wilks, 1985] Yorick Wilks. Right Attachment and Preference Semantics. *2nd Conference of the European Chapter of the A.C.L.*, 1985.

[Sekine et al., 1992] S.Sekine, S.Ananiadou, J.J.Carroll, J.Tsujii. Linguistic Knowledge Generator *submitted paper for the 14th COLING-92*, 1992