Towards a Machine Translation System with Self-Critiquing Capability

Kwangseob Shim and Yung Taek Kim

Department of Computer Engineering
Seoul National University
Seoul 151-742, Korea

Email : kshalt@krsnucc1.bitnet

Abstract

In this paper, a statistical approach to structural disambiguation of verbal phrase is discussed. Instead of hand-coded semantic knowledge that raises knowledge acquisition bottleneck problem, it is proposed to use collocations in resolving structural ambiguity. Since simple collocations do not carry information about the strength of semantic relationship, some types of ambiguity cannot be resolved by using them. They are augmented with information-theoretic concept of mutual information, so that they can reflect the strength of semantic relationship. Augmented collocations can be acquired automatically from text corpora. A new concept of confidence measure is also proposed that can be used as a good criterion to indicate the reliability of attachment. Experiments validated that the confidence measure is closely related with the accuracy of attachment. Once a threshold is set empirically, a machine translation system can have the self-critiquing capability, that is, a capability of guessing whether the attachment determined by itself is correct or not.

1. Introduction

The demand for a fully-automated high-quality machine translation system is high and growing in our information-saturated world. Many researchers have thrown their efforts into building such a system. As a result, there have been great technological advances during the last several decades. The current state of the art in machine translation is, however, not yet satisfactory, even though it is encouraging. We can enumerate several reasons for this. Among them and the most important are the knowledge acquisition bottleneck problem and the lack of self-critiquing capability.

Machine translation requires an enormous amount of knowledge. During the early years of machine translation research, the knowledge had been encoded by hand. Though a laboratory prototype of a machine translation system can be constructed in that way, it is difficult to upgrade its quality to a practical system because of the so-called knowledge acquisition bottleneck problem. The more we want our system to know, the more we have to hand-feed it. This is a very difficult, expensive and time-consuming task. Therefore, automatic knowledge acquisition has emerged as one of the immediate challenges for the development of a practical machine translation system. Interesting ideas have been proposed on automatic acquisition of knowledge from various sources such as on-line dictionaries (or machine-readable dictionaries), encyclopedias, text corpora etc [4, 5]. Clearly, this line of research is helpful not only to scale up a prototype to a practical system, but also to apply the system to another domain of discourse. The knowledge acquisition bottleneck problem seems to be conquered someday in the future.

There remains another serious problem that has been neglected so far. That is the lack of self-critiquing capability. A machine translation system does merely produce the translation of source texts, and it does not guarantee the quality of the translation. The translation may not be true to the originals (source texts), but the system does not issue any indication of ill translation or at least the possibility of ill translation. Therefore, an additional process of post-editing is assumed to be necessary if the system is to be used practically. A human reviewer should review the translation thoroughly and revise errors. This has been regarded as a matter of course because of the fact that even a human translator does not always translate accurately. In this point of view, there seems to be no differences between a human translator and a machine translator (a machine translation system). A human translator is, however, different from a machine translator in that the

human translator has the capability of guessing whether the translation is correct or not. That is, he or she has the self-critiquing capability as it is called in this paper. The human translator can put a special mark on those sentences which translation, he or she thinks, may be wrong, so that the reviewer may review only the marked sentences with special carefulness. The reviewer does not have to waste his or her time reviewing the unmarked sentences which are potentially well-translated. The total amount of time consumed in the process of translate-and-review could be reduced considerably. In the case of machine translation system without self-critiquing capability, a human reviewer should review all the sentences generated by the system because the system cannot tell potentially ill-translated sentences from well-translated sentences. Much time will be wasted in the process reviewing well-translated sentences. This can be avoided if the machine has a self-critiquing capability, that is, the capability of guessing whether the translation is correct or not.

In this paper, we will discuss how to implant the self-critiquing capability in a machine translation system. As a starting point towards a system with such a capability, a statistical approach to the resolution of structural ambiguity of verbal phrases will be discussed in the paper because structural ambiguity is the main source of poor translation in English-to-Korean machine translation. In English, phrases such as prepositional phrases, *to*-infinitives, present participles etc can be used as verb modifiers or noun modifiers without morphological change. In Korean, however, different postpositions or endings are used, depending on whether the phrases are used as verb modifiers or noun modifiers. Therefore, if structural ambiguities are not resolved in source texts, Korean texts generated by the system will not carry the accurate meaning of source texts.

Knowledge for structural disambiguation is represented in collocaitonal form and is automatically acquired from corpora. Since in some cases the structural ambiguity cannot be resolved by using simple collocations, simple collocations will be augmented with information-theoretic concept of mutual information. The concept of confidence measure is also proposed in this paper that can be used to indicate the reliability of attachment. Experiments validated the confidence measure is closely related with the accuracy of attachment. Two corpora were used in the experiments. The one is a small corpus of 51,235 words collected from IBM computer manuals. The other is scientific abstracts provided by the U.S. Department of Energy and consists of 1.6 million words. Once a threshold is set empirically, a machine translation system can have the self-critiquing

capability of guessing whether the attachment determined by itself is correct or not.

2. Structural Disambiguation Using Simple Collocations

2.1 The Definition of Simple Collocation

In a natural language, there are restrictions on words that can co-occur in a sentence with a particular word. For example, we can say *a heavy smoker* or *strong tea*, but not *a strong smoker* or *heavy tea*. This example shows that *heavy* can co-occur with *a smoker* in a sentence but not with *tea* and, similarly, *strong* with *tea* but not with *a smoker*. These restrictions can be represented by the concept of collocation. In a natual language L, the collocation of a word w is defined to be a set of words that co-occur with w in common usage of the language, and is denoted in this paper as $\boldsymbol{g}(w)$ where $\boldsymbol{g}$ indicates the grammatical function of the words in the collocation to the word w. For example, **sub**(w) and **obj**(w) represent those words that collocate with a verb w as subject and object of w, respectively. $g(w)$ can be formally stated as follows.

$$g(w) = \{\, w_i \mid w_i \in L \text{ and } w_i \text{ collocates with } w \,\}$$

Collocations thus defined are called *Simple Collocations* in this paper.

Simple collocations can be automatically acquired from text corpora. A syntactic parser produces syntactic sketches in which structural ambiguity remains unresolved. Word pairs of interest are extracted from the syntactic sketches. Only predicate-argument pairs are necessary for structural disambiguation of verbal phrase. Consider the following sentence.

(1) The CONNECT command can include a user id to identify the user.

From the syntactic sketch of the sentence, it is captured that the noun *command* is the subject of the verb *include*. Therefore, *command* is added to **sub**(*include*). Similarly, *id* is added to **obj**(*include*), and *user* to **obj**(*identify*) because they are the object of the verb *include* and *identify*, respectively. In this way, simple collocations can be collected from text

corpora. For example, a set of words that can be used as a subject of the verb *read*, that is, **sub**(*read*) = {..., *operator, person, program, programmer, user, you,* ...} is acquired from the test corpus[1].

2.2 Structural Disambiguation

Simple collocations are used in the resolution of structural ambiguity of verbal phrases, as described in this section. For example, consider the sentence (2). It is structurally ambiguous because two different interpretations are possible, according to whether the *to*-infinitive modifies the verb *enter* or the noun *command*.

(2) Operators can enter the ESPC command to read the status of all drives.

The *to*-infinitive may be attached to the noun *command* if the noun is appropriate for the subject of the verb *read*. Likewise, it may be attached to the verb *enter* if the noun *operator* is appropriate for the subject of the verb *read*. As mentioned above, **sub**(*read*) = {..., *operator, person, program, programmer, user, you,* ...} is acquired from the test corpus. This means that *operator* is appropriate for the subject of *read*, but *command* is not. Therefore, the *to*-infinitive is attached to *enter* in the case of sentence (2).

Simple collocations are used in this way for structural disambiguation of verbal phrase attachment. This approach is basically similar to the disambiguation methods that employ semantic formulas [6] or preference rules [2]. It is, however, different from those methods in that structural ambiguity is resolved by using simple collocations instead of semantic features that are hand-coded by experts. In contrast, simple collocations are not hand-coded but acquired automatically from corpora.

2.3 Problems in Using Simple Collocations for Structural Disambiguation

There are some problems in using simple collocations for structural disambiguation. From the example above, it is clear that exactly one noun should be appropriate for the subject

1 The test corpus consists of 3,539 sentences (51,235 words) collected from two volumes of IBM computer manuals.

of an ambiguous verbal phrase. Otherwise, the ambiguity cannot be resolved. Therefore, there could be two types of structural ambiguity that cannot be resolved using simple collocations. First, consider the following sentences.

(3) The storage program calls the OSMI program to store the object in the database.
(4) The program reads the Recovery table to obtain the last record processed.

Since the subject and the object of the main sentence are the same, it is impossible to decide where to attach the *to*-infinitive in sentence (3). Now, let us look at the sentence (4). According to our corpus, both *program* and *table* are members of **sub**(*obtain*). This means that they are equally appropriate for the subject of *obtain*. Notice that it cannot be determined only with simple collocations which is more appropriate for the subject of the verb *obtain*. Therefore, the ambiguity cannot be resolved using simple collocations in sentence (4), either. This is one of the two types of structural ambiguity that cannot be resolved using simple collocations. The following examples show the other type of ambiguity that cannot be resolved using them.

(5) Only users processing RESOURCE authority can acquire space to hold tables.
(6) This type of view can also limit a user's ability to access privilege information in a table.

According to the test corpus, neither the noun *user* nor the noun *space* is the member of **sub**(*hold*). This phenomenon may occur while the machine is on the way of acquiring collocational knowledge, and will vanish in the long run as the acquisition process proceeds. The fact that neither *user* nor *space* is a member of **sub**(*hold*) indicates that neither of them is appropriate for the subject of the verb *hold*. Therefore, the structural ambiguity cannot be resolved in sentence (5). Now, consider the sentence (6). Neither *type* nor *ability* is the member of **sub**(*access*). In this case, they are less expected to be in **sub**(*access*) no matter how long collocational knowledge acquisition process proceeds because they are less likely to be used as a subject of the verb *access* in common usage of English. This is quite different from the phenomenon mentioned in the case of sentence (5). The ambiguity remains unresolved in sentence (6), too.

The problem shown above arises because a simple collocation does not carry information about the strength of semantic relationship. If x is a member of $\boldsymbol{g}(w)$, then x is

semantically related with w. Otherwise, x bears no semantic relationship with w. Therefore, it is necessary to augment the concept of simple collocations, so that the augmented collocations can reflect the strength of remantic relationship. It is augmented in the following section with information-theoretic concept of mutual information.

3. Structural Disambiguation Using Augmented Collocations

3.1 Augmentation of Simple Collocations with Mutual Information

According to *Transmission of Information* [3], when two words x and y have the probabilities $P(x)$ and $P(y)$, the *Mutual Information* $I_g(x,y)$ is defined to be:

$$I_g(x,y) = \log_2 \frac{P_g(x,y)}{P(x)P(y)}$$

The subscript g indicates the grammatical function of the word x to the word y. The word probabilities $P(x)$ and $P(y)$ can be estimated by normalizing $f(x)$ and $f(y)$, the number of observations of x and y in a corpus, by N, the corpus size. Similarly, the joint probability $P_g(x,y)$ can be estimated by normalizing $f_g(x,y)$, the number of times that x is followed by y, by N [1]. Therefore, the estimate of mutual information $I_g(x,y)$ is defined to be:

$$I_g(x,y) \approx \log_2 \frac{Nf_g(x,y)}{f(x)f(y)}$$

If there is a genuine association between x and y, then $P_g(x,y)$ will be much larger than $P(x)P(y)$. By definition, $I_g(x,y)$ will be much larger than zero in this case. Similarly, if there is no interesting relationship between x and y, the joint probability $P_g(x,y)$ will be less than or almost equal to $P(x)P(y)$ and thus $I_g(x,y)$ will be less than or almost equal to zero.

Since simple collocations tell us that if x is a member of $g(w)$, then x is semantically related with w, and that otherwise x bears no semantic relationship with w, the simple

collocations can be restated as follows:

$$g(w) = \{ (w_i, r_i) \mid w_i \in L \text{ and } r_i \in \{0,1\} \}$$

$$\text{where } r_i = \begin{cases} 1 & \text{if } w_i \text{ is semantically related with } w. \\ 0 & \text{otherwise.} \end{cases}$$

By the definition, r_i could be 0 or 1 according to whether w_i is semantically related with w or not. In general, however, it is impossible to clear-cut words into two classes like that. It is often the case that we can say w is semantically more closely related with w_i than with w_j. Therefore, the simple collocation should be augmented so that $r_i > r_j$ if w is semantically more closely related with w_i than with w_j. This can be formally stated as follows:

$$g'(w) = \{ (w_i, r_i) \mid w_i \in L \text{ and } r_i \in \mathbb{R} \}$$

$$\text{where } r_i = I_{g'}(w_i, w) \text{ and } \mathbb{R} \text{ is a set of real numbers.}$$

Collocations thus defined are called *Augmented Collocaitons* in this paper.

3.2 The Acquisition of Augmented Collocations

The mutual information can be calculated from corpora. The basic assumption behind this calculation is that words are distributed at random in corpora. For the calculation, it is required to obtain the word and joint frequencies and the corpus size. The same syntactic parser as mentioned in Section 2 is used in obtaining them. It is trivial to obtain the word frequencies $f(x)$ and $f(y)$; just to count the occurrences of the word x and y in corpora. Here is an example of how to obtain joint frequency $f_g(x,y)$ for subject/verb, verb/object and word/*to*-infinitive pairs. Again, consider the sentence (4) in Section 2. It is shown here again for convenience.

(4) The program reads the Recovery table to obtain the last record processed.

It is evident to increase f_{sub}(*program,read*) by one because *program* is the subject of *read*.

Similarly, it is also evident to increase f_{obj}(*table,read*) and f_{obj}(*record,obtain*) by one because *table* and *record* are the object of *read* and *obtain*, respectively. It would be problematic to determine which of f_{inf}(*read,to/to*) and f_{inf}(*table,to/to*) should be increased by one, since syntactic sketches where structural ambiguity remains unresolved are used in calculating the mutual information. The problem is solved simply by increasing both f_{inf}(*read,to/to*) and f_{inf}(*table,to/to*) by one. Increasing f_{inf}(*table,to/to*) may be thought wrong in this particular example, but it is less harmful on the whole because mutual information is calculated from corpora, not from merely a single sentence. Since words are assumed to be distributed at random in corpora, the frequency of those word pairs that have no interesting relationship will be much lower than that of those words pairs that have genuine semantic relationship. Therefore, the statistics will be saturated and become stable ultimately. In this way, the joint frequencies are obtained from corpora, and then mutual information is calculated by the equation given above. Table 1 shows mutual information for some word pairs calculated from our corpus.

x	y	$I_{sub}(x,y)$	x	y	$I_{inf}(x,y)$
program	read	4.42	read	to/to	3.05
program	obtain	4.66	table	to/to	-0.72
table	obtain	1.45	acquire	to/to	2.52
user	hold	?	space	to/to	2.13
space	hold	?	ability	to/to	6.22

Table 1. Mutual Information : subject/verb and word/*to*-infinitive pairs

3.3 Structural Disambiguation

When two words x and y are collocatable, mutual information for the word pair shows how strongly they collocate with each other. By definition, the bigger $I_g(x,y)$ represents the stronger semantic relationship between x and y. Therefore, when $I_g(x,z) > I_g(y,z)$, the semantic relationship of x and z is stronger than that of y and z. Suppose that it is ambiguous whether phrase z is attached to phrase x or y, and $I_g(x,y) > I_g(y,z)$. It is clear that the probability that the attachment of z to x is correct will be greater than the

probability that the attachment of z to y is correct. Therefore, it is reasonable to attach z to x in this case. Of course, this attachment might be wrong, but the probability would be relatively low.

Let us give an example of using augmented collocations for structural disambiguation. Consider again the sentence (4) in Section 2. As we have shown in the section, both *program* and *table* are members of **sub**(*obtain*). This means that both of them are equally appropriate for the subject of *obtain*. It is because simple collocations do not reflect the strength of semantic relationship. Therefore, the ambiguity could not be resolved using them. Now, notice that $I_{sub}(program,obtain) = 4.66 > I_{sub}(table,obtain) = 1.45$. This indicates that *program* is statistically more appropriate for the subject of *obtain* than *table* is. Therefore, it is reasonable to attach the *to*-infinitive to the verb *read* in this case. Now, consider the sentence (5) in Section 2. It is shown below for convenience.

(5) Only users processing RESOURCE authority can acquire space to hold tables.

Since neither *user* nor *space* is a member of **sub**(*hold*), the mutual information $I_{sub}(user,hold)$ and $I_{sub}(space,hold)$ can not be defined. Thus, structural ambiguity cannot be resolved using mutual information for subject/verb pairs. In this case, the mutual information of word/*to*-infinitive pairs are used, instead. The mutual information $I_{inf}(w,to/to)$ indicates how often the word w co-occurs with *to*-infinitives. An attachment is decided on the assumption that the probability of the attachment being correct will be greater if *to*-infinitive is attached to a word that co-occurs more often with *to*-infinitive. Since $I_{inf}(acquire,to/to) = 2.52 > I_{inf}(space,to/to) = 2.13$, the *to*-infinitive is attached to the verb *acquire*. Notice that the attachment is incorrect. This problem will be discussed again in the following section.

4. Confidence Measure and its Relationship with the Accuracy of Structural Disambiguation

4.1 The Definition of Confidence Measure

As we have described in Section 3, structural ambiguity can be resolved by comparing the

absolute values of mutual information. Now, let us think about their relativity. Suppose that $I_g(x,z) - I_g(y,z) \gg 0$ and $I_g(x',z') - I_g(y',z') \approx +0$. Although z and z' are attached to x and x', respectively, the possibility of the attachment being correct would be quite different from each other. There is a great possibility of correct attachment in the former case whereas there is a bare possibility in the latter case. That is, the comparison of relative values of mutual information indicates the degree of confidence in the attachment. This is a logical consequence of the fact that mutual information reflects the strength of semantic relationship. From this point of view, the new concept of *Confidence Measure* is defined.

When it is ambiguous whether phrase z is attached to phrase x or y, the ambiguity is resoloved as described in Section 3. The degree of confidence in the attachment is indicated by *Confidence Measure* $C(x,y,z)$ that is defined to be

$$C(x,y,z) = | I_g(x,z) - I_g(y,z) |$$

The bigger confidence measure indicates the more reliable attachment. As confidence measure approaches to zero, the attachment becomes less reliable. An example will clear this point. Let us return to the sentences (4) and (5) we have discussed in Section 3. In the case of sentence (4), the *to*-infinitive was attached to *read* because $I_{sub}(program,obtain) = 4.66 > I_{sub}(table,obtain) = 1.45$. In the case of sentence (5), the *to*-infinitive was attached to *acquire* because $I_{inf}(acquire,to/to) = 2.52 > I_{inf}(space,to/to) = 2.13$. Now, let us calculate the confidence measure for both cases. In the former case, $C(program,table,obtain) = 4.66 - 1.45 = 3.21$. In the latter case, $C(acquire,space,to/to) = 2.52 - 2.13 = 0.39$. These figures indicate that the probability of the attachment being correct is high in the former case whereas it is low in the latter case. The relatively low confidence measure in the latter case indicates that the attachment may be wrong and it is actually wrong as we have mentioned at the end of Section 3.

4.2 Experimental Results

We made an early experiment to verify the relationship between confidence measure and the accuracy of attachment. A total of 99 sentences were sampled from a test corpus that

consists of 51,235 words collected from IBM computer manuals. Structural ambiguities were resolved using augmented collocations acquired from the corpus. 71 sentences were disambiguated correctly. The overall accuracy was 72%. The confidence measure varied from 0.0 to 7.6. Table 2 shows the relationship between confidence measure and the accuracy of attachment. The accuracy increases as confidence measure grows.

	Correct	Incorrect	Accuracy
$0.0 \le C < 1.0$	15	13	54%
$1.0 \le C < 2.0$	8	7	53%
$2.0 \le C < 3.0$	18	6	75%
$3.0 \le C < 4.0$	15	2	88%
$4.0 \le C$	15	0	100%

Table 2. Confidence Measure and Accuracy : the early experiment

The accuracy increases rapidly when confidence measure varies from 2.0 to 3.0. We further examined on this range the relationship between confidence measure and the accuracy. There are all 24 sentences on the range. Among them, 18 sentences are disambiguated correctly and the 6 sentences incorrectly. In the former case, confidence measures are uniformly distributed on the range from 2.0 to 3.0. In the latter case, however, they are clustered on a narrow range from 2.0 to 2.1. The accuracy increases very rapidly as confidence measure crosses this value, that is, a threshold. Table 3 shows the relationship between confidence measure and the accuracy of attachment.

	Correct	Incorrect	Accuracy
$0.0 \le C < 2.1$	24	26	48%
$2.1 \le C$	47	2	96%

Table 3. Threshold and Accuracy

When the confidence measure of an attachment is greater than the threshold, in our

case 2.1, the attachment will be correct with the probability of 0.96. Only one attachment will be incorrect out of 25. When the confidence measure is lower than the threshold, to the contrary, the attachment will be correct with the probability of 0.48. Every other attachment will be wrong. This experiment shows that confidence measure can be used as a reliable criterion for deciding whether an attachment is correct or not.

We made a second experiment with the DOE corpus of 1.6 million words[2]. A total of 613 sentences were sampled from the corpus. The result was similar to that of the early experiment. The overall accuracy was 72%; among 613 sample sentences, 443 sentences were disambiguated correctly. Table 4 shows the relationship between cofidence measure and the accuracy of attachment. It is interesting to compare Table 2 and Table 4. The accuracy was 91% when the confidence measure was greater than 3.0.

	Correct	Incorrect	Accuracy
$0.0 \le C < 1.0$	44	56	44%
$1.0 \le C < 2.0$	57	51	53%
$2.0 \le C < 3.0$	89	39	70%
$3.0 \le C < 4.0$	99	17	85%
$4.0 \le C < 5.0$	103	7	94%
$5.0 \le C$	51	0	100%

Table 4. Confidence Measure and Accuracy : the 2nd experiment

5. Confidence Measure in Supervised Learning Model

By introducing the concept of confidence measure and by setting a threshold of confidence measure, a machine translation system is endowed with the self-critiquing capability of guessing whether the attachment determined by itself is correct or not. Therefore, the system will be able to issue a warning signal when the confidence measure is lower than the threshold, but it still has a problem. The problem is that although the system can detece potential errors in attachment, the system cannot correct them by itself. It will issue

2 The DOE corpus consists of scientific abstracts provided by the U.S. Department of Energy.

the same warning signal on the same errors. It could be prevented if the knowledge base is improved. In this section, a simple model of supervised learning to cope with the problem will be discussed. A supervised learning model is a learning model that involves a human teacher in the learning process. Figure 1 shows the learning model. The self-critiquing capability plays an important role in this model.

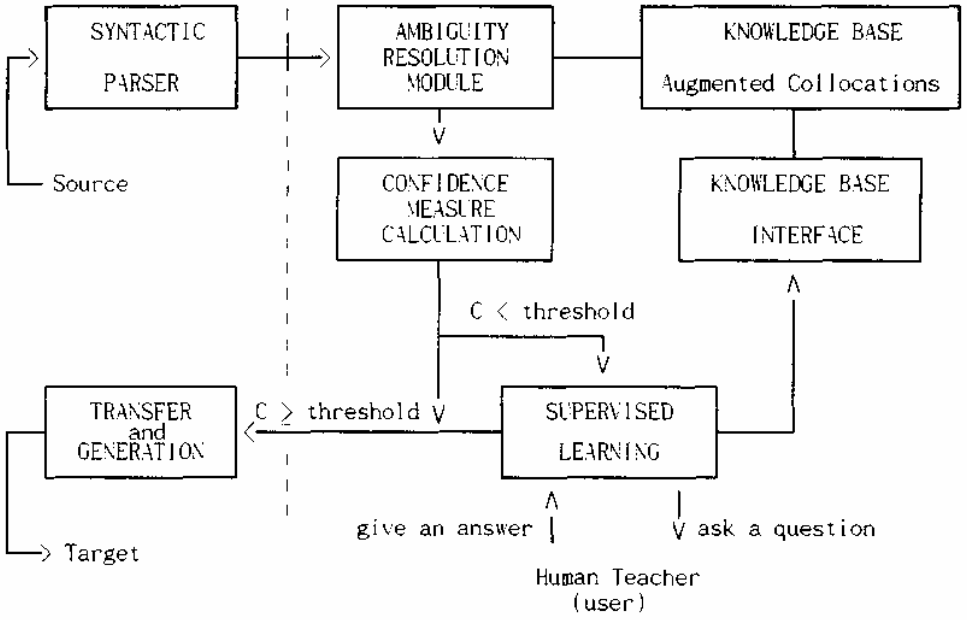

Figure 1. The Supervised Learning Model

Initially, the knowledge necessary for structural disambiguation, that is, augmented collocations, are automatically acquired from text corpora and then stored in KNOWLEDGE BASE. SYNTACTIC PARSER produces parse trees in which structural ambiguity remains unresolved. They are resolved by AMBIGUITY RESOLUTION MODULE. If the probability of the resolution being correct is high, that is, the confidence measure is greater or equal to the threshold, the potentially ambiguity-free parse tree is input to the next stage, say, TRANSFER and GENERATION module. Otherwise, SUPERVISED LEARNING module is activated. The module issues a question about whether the resolution is correct or not. The human teacher answers the question with "yes" or "no". If the teacher answers with "no", the learning module will issue a request to KNOWLEDGE BASE INTERFACE to revise

the mutual information portion of augmented collocations stored in KNOWLEDGE BASE. Direct revision of mutual information is dangerous because it may cause the system to behave in an unpredictable way. Therefore, the revised mutual information should be stored in KNOWLEDGE BASE as an exceptional case. For example, suppose that $I_g(x,z) > I_g(y,z)$ and the attachment of z to x is incorrect. The mutual information $I_g(y,z)$ is revised to $I_g(x,z) + \varepsilon$ and stored in KNOWLEDGE BASE with the additional access key x. ε is a very small positive real number. The system will attempt first to access the revised mutual information. It will access the unrevised mutual information acquired from corpora only in case that the revised mutual information is not available.

Since the knowledge sources cannot be expected to be complete in all cases, it is necessary to enhance the quality of knowledge acquired. Moreover, the knowledge could be temporarily incomplete while the acquisition process proceeds. The supervised learning model is proposed to cope with this situation. The KNOWLEDGE BASE is updated by the automatic acquisition process and refined by man-machine interaction as shown above.

6. Conclusion

In this paper, a statistical approach to structural disambiguation of verbal phrases was discussed. Knowledge for structural disambiguation is represented in collocational form and is automatically acquired from corpora. Since in some cases the structural ambiguity cannot be resolved by using simple collocations, we have augmented simple collocations with information-theoretic concept of mutual information. We have also proposed the concept of confidence measure that can be used as a good criterion for deciding whether an attachment is correct or not. An experiment validated the confidence measure is closely related with the accuracy of attachment. Once a threshold is set empirically, a machine translation system can have the self-critiquing capability of guessing whether the attachment determined by itself is corect or not.

Since ambiguities are inherent in a natural language, decision-making is very important in machine translation. Decision-making itself would be easy if the knowledge were complete. In reality, however, knowledge is not complete whether it is encoded by hand or constructed automatically. Therefore, a machine translation system that uses the incomplete knowledge may generate ill-translated sentences. A machine translation system

with the self-critiquing capability could distinguish ill-translated sentences from those sentences that are well-translated. The capability is important to a machine translation system in two points. First, the total amount of time consumed in the process of translate-and-review can be reduced so that the benifit of using a machine translation system increases. Second, we can improve the knowledge base semi-automatically and effectively. From this point of view, we argue that self-critiquing capability should be one of the characteristics of a machine translation system in next generation.

References

[1] Kenneth Ward Church and Patrick Hanks, "Word Association Norms, Mutual Information, and Lexicography," Computational Linguistics, Vol.16, No.1, pp.22-29 (1990).

[2] K. Dahlgren and J. McDowell, "Using Commonsense Knowledge to Disambiguate Prepositional Phrase Modifiers," Proceedings of AAAI-86, pp.589-593 (1986).

[3] R. Fano, *Transmission of Information: A statistical Theory of Communications*, MIT Press (1961).

[4] Donald Hindle and Mats Rooth, "Structural Ambiguity and Lexical Relations," Proceedings of 29th Annual Meeting of the ACL, pp.229-236 (1991).

[5] Takehito Utsuro, Yuji Matsumoto and Makoto Nagao, "Lexical Knowledge Acquisition from Bilingual Corpora," Proceedings of COLING-92, pp.581-587 (1992).

[6] Y. Wilks, X. Huang and D. Fass, "Syntax, Preference and Right Attachment," Proceedings of IJCAI-85, pp.779-784 (1985).