Two Methods for Learning ALT-J/E Translation Rules from Examples and a Semantic Hierarchy

Hussein Almuallim

Info. and Computer Science Dept.
King Fahd University of
Petroleum and Minerals
Dhahran 31261, Saudi Arabia

Yasuhiro Akiba Takefumi Yamazaki
Akio Yokoo Shigeo Kaneda

NTT Communication Science Labs.
1-2356, Take, Yokosuka-shi
Kanagawa-ken 238-03
Japan

Abstract

This paper presents our work towards the automatic acquisition of translation rules from Japanese-English translation examples for NTT's ALT-J/E machine translation system. We apply two machine learning algorithms : Haussler's algorithm for learning internal disjunctive concept and Quinlan's ID3 algorithm. Experimental results show that our approach yields rules that are highly accurate compared to the manually created rules.

1 Introduction

A critical issue in AI research is to overcome the knowledge acquisition bottleneck in knowledge-based systems. As a knowledge base is expanded, adding more knowledge and fixing previous erroneous knowledge become increasingly costly. Moreover, maintaining the integrity of large knowledge bases has proven to be a very challenging task.

A widely proposed approach to deal with the knowledge acquisition bottleneck is to employ some learning mechanism to extract the desired knowledge automatically or semi-automatically from actual cases or examples [Buchanan & Wilkins 1993]. The validity of this approach is becoming more evident as various machine-learning-based knowledge acquisition tools for real-world domains are being reported [Kim & Moldovan 1993, Porter et al. 1990, Sato 1991a, Sato 1991b, Utsuro et al. 1992, Wilkins 1990].

ALT-J/E, which is an experimental Japanese-English translation system developed at Nippon Telegraph and Telephone Corporation (NTT), is one example of a large knowledge-based system in which solutions to the knowledge acquisition bottleneck are definitely needed. One major component of this system is its huge collection of *translation rules*. Each of these rules associates a Japanese sentence pattern with an appropriate English pattern. To translate a Japanese sentence into English, ALT-J/E looks for the rule whose Japanese pattern matches the sentence best, and then uses the English pattern of that rule for translation.

So far, ALT-J/E translation rules have been composed manually by extensively trained human experts. To qualify for this job, an expert must not only master both English and Japanese, but also be very familiar with various components of the system. Each time the rules are expanded or altered, the new set of rules must then be "debugged" using a collection of test cases. Usually, several iterations are needed to arrive at translation rules of acceptable quality.

Creating new translation rules as well as refining existing ones have proven to be extremely difficult and time-consuming because these tasks require considering a huge space of possible combinations (rules in ALT-J/E are expressed in terms of as much as 3000 "semantic categories"). The high costs involved make the manual creation of ALT-J/E's translation rules impractical. Indeed, in spite of the vast amount of resources spent on building the current rules of ALT-J/E, faults in these rules are still detected from time to time, making system maintenance a continuous requirement.

The aim of this work is to make ALT-J/E's translation rules less costly and more reliable through the use of inductive machine learning techniques. Careful examination of the manual process which has been followed so far by ALT-J/E's experts for building translation rules reveals that most of the effort is spent on figuring out the condition part of the rules (that is, the Japanese patterns). Therefore, we propose the use of inductive machine learning algorithms to learn these conditions from examples of Japanese sentences and their English translations. Under this machine learning approach, the user is relieved from exploring the huge space of alternatives she/he has to consider when constructing translation rules manually from scratch – a job which only extensively trained experts can perform. The task is now turned into a search for some reasonable rules that explain the given training examples, where the search is handled automatically by a learning algorithm. This not only saves the user's time, but also makes it unnecessary for the user to be an expert of the ALT-J/E system. Moreover, this approach significantly reduces the "subjectivity" of the rules since the intervention of human experts is minimized. This is particularly important because the immense number of translation rules (currently over 10,000) requires employing a team of experts over an extended period of time.

Two learning methods are investigated in this paper. Experiments show that the rules learned by these methods are very close to the rules manually composed by human experts. In most cases, given a reasonable number of training examples, the employed methods are able to find rules that are more than 90% accurate when compared to the manually composed rules.

The rest of this document is organized as follows. We begin in Section 2 by a brief overview of the ALT-J/E Japanese-English translation system. In Section 3, we discuss some of the problems that arise when the translation rules of ALT-J/E are composed manually by human experts. Then, we propose in Section 4 an alternative approach based on machine learning techniques. In Section 5, we describe the inductive learning methods used, followed by an experimental evaluation of these methods in Section 6. Finally, conclusion remarks are stated in Section 7.

2 ALT-J/E: A Brief Overview

ALT-J/E, the **Automatic Language Translator: Japanese to English**, is one of the most advanced and well-recognized systems for translating Japanese to English. It is the largest such system in terms of the amount of knowledge it comprises. In this work, we are concerned with the following components of the ALT-J/E system:

1. The Semantic Hierarchy,
2. The Semantic Dictionary, and
3. The Translation Rules.

We briefly describe each of these components below. For more details about the ALT-J/E system, we refer the reader to [Ikehara et al. 1989, Ikehara et al. 1990, Ikehara et al. 1991].

As shown in Figure 1, the **Semantic Hierarchy** is a sort of concept thesaurus represented as a tree structure in which each node is called a *semantic category*, or a *category* for simplicity. Edges in this structure represent "is-a" relations among the categories. For example, "Agents" and "People" (see Figure 1) are both categories. The edge between these two categories indicates that any instance of "People" is also an instance of "Agents". The current version of ALT-J/E's Semantic Hierarchy is 12 levels deep and has about 3000 nodes. The **Semantic Dictionary** maps each Japanese noun to its appropriate semantic categories. For example, the Semantic Dictionary states that the noun 鶏 (niwatori), which means "chicken" or "hen" in English, is an instance of the categories "Meat" and "Birds".

The **Translation Rules** in ALT-J/E associate Japanese patterns with English patterns. Currently, ALT-J/E uses roughly 10,000 of these rules.[1] As Figure 2 shows, each translation rule has a Japanese pattern as its left-hand side and an English pattern as its right-hand side. For example, the first rule in this figure basically says that if the Japanese verb in a sentence is 焼く (yaku), its subject is an instance of "People", and its object is an instance of "Bread" or "Cake", then the following English pattern is to be used:

Subject "bake" Object.

Note that in this case the Japanese verb 焼く (yaku) is translated into the English verb "bake". This same Japanese verb can also be translated into the English verbs "roast", "broil", "cremate" or "burn", depending on the context. These cases are handled by the four other rules given in Figure 2.

Translation rules are meant only to handle basic sentences that contain just a single Japanese verb. Such sentences are called "simple sentences."[2] To translate a complex sentence, ALT-J/E does various kinds of pre- and post-processing. Roughly speaking, the given complex sentence is first broken into a collection of simple sentences in the pre-processing phase. Then, the English translations of these are combined together in the post-processing phase to give the final translation of the complex sentence.

To translate a simple sentence, ALT-J/E looks for the most appropriate translation rule to use. Based on the verb of the sentence, the system considers as candidates all those translation rules that have this verb on their left-hand side. The English pattern of the rule whose Japanese pattern matches the sentence best is then used to generate the desired English translation.

As shown in Figure 2, the Japanese patterns are expressed using the variables $N_1, N_2, \cdots$, etc., which represent various components of a Japanese sentence, such as the subject, the object, etc.[3] The "degree of matching" between a Japanese pattern and a sentence is based on how well the values of these variables for the given sentence match those categories required by the Japanese pattern. The Semantic Dic-

[1] In fact, ALT-J/E has three different kinds of translation rules: (i) the semantic pattern transfer rules (roughly 10,000 rules), (ii) the idiomatic expression transfer rules (about 5,000 rules), and (iii) the general transfer rules. We use the term "Translation Rules" here to refer to the semantic pattern transfer rules. These form the majority of the rules, and they are the most frequently used by ALT-J/E.

[2] The term "simple sentence" is a direct translation of 単文 (tanbun) in Japanese.

[3] To be precise, Japanese sentences are usually parsed into a set of components (called が-格, を-格, に-格, etc.) that are quite different from those used in English. Using "subject" and "object" here is only meant to ease the discussion for English readers.

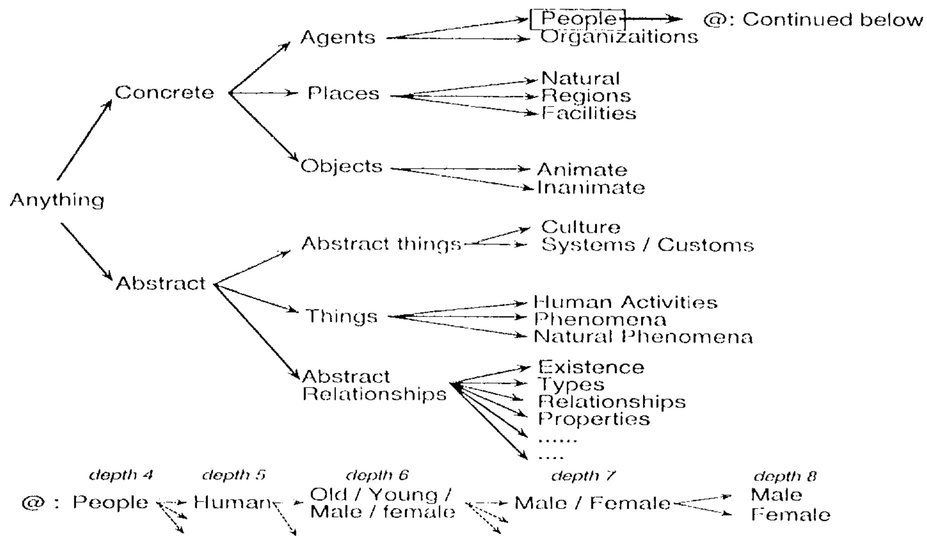

Figure 1: The upper levels of the Semantic Hierarchy in ALT-J/E.

IF			THEN		
	J-Verb	= "焼く (yaku)"		Subj	= N_1
	N_1 (Subj)	≡ "People"		E-Verb	= "bake"
	N_2 (Obj)	≡ "Bread" or "Cake"		Obj	= N_2
IF			THEN		
	J-Verb	= "焼く (yaku)"		Subj	= N_1
	N_1 (Subj)	≡ "People"		E-Verb	= "roast"
	N_2 (Obj)	≡ "Meat"		Obj	= N_2
IF			THEN		
	J-Verb	= "焼く (yaku)"		Subj	= N_1
	N_1 (Subj)	≡ "People"		E-Verb	= "broil"
	N_2 (Obj)	≡ "Fish" or "Seafood"		Obj	= N_2
IF			THEN		
	J-Verb	= "焼く (yaku)"		Subj	= N_1
	N_1 (Subj)	≡ "Agents"		E-Verb	= "cremate"
	N_2 (Obj)	≡ "People" or "Animals"		Obj	= N_2
IF			THEN		
	J-Verb	= "焼く (yaku)"		Subj	= N_1
	N_1 (Subj)	≡ "Agents" or "Machines"		E-Verb	= "burn"
	N_2 (Obj)	≡ "Places" or "Objects" or "Locations"		Obj	= N_2

Figure 2: Translation rules for the Japanese verb 焼く (yaku). These rules are composed manually by human experts. " ≡ " indicates "an instance of".

tionary is used during the matching process to determine whether or not a given noun is an instance of a certain category.

3 Shortcomings of the Manual Approach

Translation rules in the ALT-J/E system have so far been composed manually by human experts. However, due to the high cost-per-rule, and because of the huge number of translation rules needed for ALT-J/E to carry out a reasonable translation job, the manual approach has been concluded by the developers of ALT-J/E to be impractical. In particular, the following problems have been reported:

- Building and maintaining the translation rules require a great deal of expertise. To qualify for this task, skillful experts are required not only to master both Japanese and English, but also to be fully familiar with ALT-J/E's large Semantic Hierarchy and to understand the overall process of the system. Such qualifications are costly and involve extensive training.

- In spite of the vast amount of resources spent on building the current rules of ALT-J/E by human experts, faults are still detected from time to time, making the maintenance of the system a continuous requirement.

- The translation rules are not quite *concrete* and vary depending on the expert. Rules constructed by one expert are not easy for another expert to understand and modify. This makes the maintenance process more difficult and makes it hard to substitute an expert by another.

- An important objective is to build specialized versions of ALT-J/E to be used in specific application domains. The manual approach is obviously unrealistic since it involves more training of the human experts with respect to the target application domain, and because this process has to be repeated for every new domain.

- One of the problems facing the designers of ALT-J/E is the refinement of the Semantic Hierarchy. Whenever this structure is altered, the translation rules must also be revised to reflect the change. Such revision is extremely troublesome and error-prone if it is done manually.

4 A Machine Learning Approach

The problems we have just listed regarding the manual construction of ALT-J/E's translation rules are largely solved if the process can be automated. An attractive approach to this problem is to resort to inductive machine learning techniques to extract the desired translation rules from examples of Japanese sentences and their English translations. At the current stage, however, learning translation rules fully automatically from examples alone seems to be too challenging. A more realistic goal is to minimize — rather than to totally eliminate — the intervention of human experts in the rule aquisition process. Thus, our current objective is to concentrate on automating the most difficult and time-consuming parts of the manual procedure.

The goal of the present work is to learn what we call "partial translation rules". A partial translation rule consists of the left-hand side along with the English verb of the right-hand side of a translation rule. In other words, the only difference between a translation rule and a partial translation rule is that the latter has only an English verb rather than a full English pattern as its right-hand side.

Constructing a partial translation rule is the most difficult part of constructing a translation rule. Indeed, turning a partial rule into a complete one is a relatively easy task that can be done by a human operator with moderate knowledge of English and Japanese.

5 Learning Task and Methods

In this work, we investigate two different inductive learning algorithms. Before talking about these algorithms, we will first make the learning task more precise and shed some light on the difficulties that distinguish it from other previously studied learning tasks.

5.1 The Learning Task

The job of a learning algorithm in our setting is to construct partial translation rules. For a given Japanese verb *J-verb* and a possible English translation *E-verb$_i$* of that verb, the algorithm has to find the appropriate condition(s) that should hold in the context in order to map *J-verb* to *E-verb$_i$*.

As an example, consider the Japanese verb 使う (tsukau). This verb corresponds to the English verbs "use", "spend" and "employ". The choice among these English verbs depends mostly on the object of the sentence. For example, if the object is an instance of "Asset" or "Time", then "spend" is appropriate. Thus, a rough rule for mapping 使う (tsukau) to "spend" may look like

IF $\boxed{\text{J-VERB} = \text{使う}}$
and $\boxed{\text{OBJECT } \textit{is an instance of} \text{ "Time" or "Asset"}}$
THEN E-VERB = spend.

We seek to learn this kind of rules from examples of Japanese sentences and their English translations, such as the following pair:

⟨ 王女が金を使う, The princess spends money ⟩.

After parsing (which is carried out by ALT-J/E's parser), the above example gives the following pair:

⟨ [J-VERB = 使う , SUBJECT = oujyo.
OBJECT = kane], E-VERB = spend ⟩.

By looking up the Semantic Dictionary of ALT-J/E, the possible semantic categories for oujyo are "Noble Person", "Daughter" and "Female", and those for kane are "Asset", "Metal", "Day" and "Medal". Thus, this example is finally given to the learning algorithm in the following form:

⟨ [SUBJECT ≡ { Noble Person, Daughter, Female },
OBJECT ≡ { Asset, Metal, Day, Medal }] ,
E-VERB = spend ⟩,

where $N \equiv S$ indicates that the sentence component N is an instance of each category $s \in S$. The general format of the training examples is as follows:

$$\begin{aligned} \langle [\quad & N_1 \equiv \{a_1, a_2, \cdots\}, \\ & N_2 \equiv \{b_1, b_2, \cdots\}, \cdots \\ & N_n \equiv \{c_1, c_2, \cdots\}], \textit{E-Verb}\rangle \end{aligned} \tag{1}$$

where each N_i represents a component of the sentence (subject, object, etc.), and each a_i, b_i, and c_i is a semantic category.

From the viewpoint of machine learning research, the above learning task is interesting/challenging from two perspectives:

- **Huge amount of background knowledge:** To be appropriate for our learning task, the learning algorithm must effectively utilize ALT-J/E's large Semantic Hierarchy. This requirement of being capable of exploiting such a huge amount of background knowledge disqualifies most of the known inductive learning algorithms from directly being used in our domain.
- **Ambiguity of the training examples:** Unlike most known learning domains, the training examples in our setting (as given in Eq. (1)) are *ambiguous* in the sense that each of the variables (SUBJECT, OBJECT, etc.) is assigned multiple values rather than a single value. Focusing on the relevant values (that is, the values that contributed to the choice of the English verb) is an extra challenge to the learner in our domain.

To deal with the above learning problem, we investigated two approaches. One is based on a theoretical algorithm introduced by Haussler for learning internal disjunctive concepts, and the other on the well-known ID3 algorithm of Quinlan.

5.2 Haussler's algorithm for learning internal disjunctive expressions

In our first approach, we represent the conditions of the learned partial translation rules as *internal disjunctive expressions*, and employ an algorithm given by Haussler for learning concepts expressed in this syntax. Haussler's algorithm enjoys many advantages. First, it has been analytically proven to be quite efficient both in terms of time and the number of examples needed for learning. Second, the algorithm is capable of explicitly utilizing the background knowledge represented by the Semantic Hierarchy. Moreover, the language used by human experts to construct ALT-J/E's rules is quite similar to internal disjunctive expressions, suggesting the appropriateness of this algoeithm's bias. Haussler's algorithm, on the other hand, suffers the important shortcoming (within our setting) that it is not capable of learning from ambiguous examples. In order to be able to use the algorithm for our task, the ambiguity has to be explicitly removed from all the training examples. Of course, this approach is not desirable because it requires some intervention by a human expert and because there are no guarantees that disambiguation is done in a perfect manner.

5.3 Quinlan's ID3

Our second approach is based on the ID3 algorithm introduced by Quinlan in [Quinlan 1986]. As it is, ID3 is not able to utilize the background knowledge of our domain, nor is it capable of dealing with ambiguous training examples of the form given by Eq. (1). It is clearly inappropriate to treat $N_1, N_2 \cdots$ as multi-valued variables, which is the most common way of using ID3. This is because of the huge number of values these variables can take, and also because we need to exploit the background knowledge represented by the Semantic Hierarchy.

To be able to use ID3 in our domain, we transform the training examples into a new representation that can be handled by ID3. The transformation we propose is done in a way such that the relevant information from the the Semantic Hierarchy are included in the newly represented examples, and, at the same time, these newly represented examples still reflect the ambiguity present in the original examples.

Our transformation method is described as follows: Let A be the set of all the categories that appeared in the training examples, and their ancestors. For every $c \in A$, we define a binary feature as a test of the form

Is N_i an instance of c?

For a training example

$$\langle [N_1 \equiv S_1, \cdots N_i \equiv S_i, \cdots N_n \equiv S_n], \textit{E-Verb}\rangle,$$

we let the outcome of the above test be *true* if and only if there exists some $s \in S_i$ such that s is an ancestor of c in the Semantic Hierarchy, or c itself. Using these features, we convert each of the training examples into a new pair $\langle V, \textit{E-Verb}\,\rangle$ where V is a vector of bits each representing the outcome of the corresponding feature for the given training example.

Given the above definition of the binary features, the new pairs ⟨V, *E-Verb* ⟩ include all the necessary background knowledge obtained form the Semantic Hierarchy, and also reflect the ambiguity of the original training examples. In other words, the above transformation can be seen as "compiling" the information of the original ambiguous training examples along with the necessary parts of the Semantic Hierarchy into a format that is ready to be processed by ID3 (or in fact, by many other feature-based learning algorithms).

Note that if we create a feature for every semantic category c and every sentence component N_i, then the total number of features will become infeasibly large (many thousands). However, what we need is only to consider those categories that appeared in the training data, and their ancestors (the set A above). In our experiments, this results in a reasonable number of features (one to two hundred). This is because the number of examples is limited and also because of the rather "tilted" distribution of what categories can naturally appear as a certain component of a sentence for a given verb. (Eg. the object of the verb 飲む (nomu), which roughly means to "drink", can not be just anything!)

The most important advantage of the above approach is that it can be applied to ambiguous training examples as they are, without the need to remove the ambiguity explicitly as we did with Haussler's algorithm. Another advantage of using ID3 is that we do not need to break our learning task into binary class learning problems since ID3 is capable of learning multi-class learning concepts.

6 Experimental Work

The goal of the experiments reported here is to evaluate the quality of the partial translation rules learned by the two learning methods we have just described. The comparison includes the following three settings:

1. Using Haussler's algorithm to learn from training examples after removing the ambiguity.

2. Using ID3 to learn from training examples after removing the ambiguity and performing the transformation given in the Subsection 5.3.

3. Using ID3 to learn from training examples after performing the transformation given in the Subsection 5.3, but without removing the ambiguity.

In a sense, the first setting represents the best we can do in the absence of the ambiguity since Haussler's algorithm does a good job in exploiting the background knowledge from the Semantic Hierarchy. Comparing Setting 2 with Setting 1 tells us how successful our transformation of the training examples is in letting ID3 make use of the available background knowledge. Finally, comparing Setting 3 with Setting 2 tells us how successful our transformation is in letting ID3 learn directly from ambiguous training examples.

The experiments were done for six different Japanese verbs. Table 1 shows a list of these verbs, along with the number of training examples used, and the accuracy levels obtained by each method. In the table, "Haussler", "ID3–NA" and "ID3–A" denote Setting 1, Setting 2 and Setting 3, respectively. The accuracy was estimated using the leave-one-out cross-validation method[4], and assuming that the rules composed manually by human experts are perfect (that is, we are measuring how close the learned rules are to those composed manually).

The performance levels of both Haussler's algorithm and ID3 when learning from unambiguous examples are quite similar in spite of the fact that each algorithm implements a different bias and has a completely different way of exploiting the background knowledge. Comparing the performance of ID3 in the two cases of learning from ambiguous and unambiguous examples, ambiguity is not harmful to ID3's performance in most cases. In fact, for some of the verbs, the performance is even better when ambiguity is present. This suggests that the approach we have chosen to deal with ambiguity is effective for our task, and that explicit removal of ambiguity is not an attractive strategy since it is not easy to do, and since it does not greatly improve the accuracy anyway.

The most important point here is that the observed accuracy of both the ID3 algorithm and Haussler's algorithm is satisfactorily high overall in spite of the limited number of the training examples used. Such a high level of accuracy strongly indicates that the use of these algorithms will provide significant aid in the construction of ALT-J/E's translation rules.

7 Conclusion

This paper reported our work towards the acquisition of Japanese-English translation rules through the use of inductive machine learning techniques. Two approaches were investigated. The first approach is based on a theoretically-founded algorithm given by Haussler for learning internal disjunctive concepts. This algorithm has the advantage that it is tailored to utilize background knowledge of the kind available in our domain. We found, however, no obvious way to make this algorithm learn directly from ambiguous training examples, and thus, ambiguity was explicitly removed from the training examples in order to use this algorithm. Our second approach is based on the ID3 algorithm. As it is, ID3 is not able to utilize the background knowledge of our domain, nor is it capable of dealing with ambiguous training exam-

[4] Examples are excluded from the training set one at a time. The rule learned from the rest of the examples is then used to predict the class of the removed example. This was repeated for all the examples, and the percentage of correct classification is reported.

Table 1: Experimental results on six Japanese verbs. Numbers show the accuracy per-cent, estimated using the leave-one-out cross-validation method. ID3-NA indicates using ID3 with the ambiguity removed from the training examples. ID3-A indicates using ID3 to learn from ambiguous training examples.

Japanese Verb	*English Verbs*	*No. of Exs.*	*Accuracy %* Haussler	ID3 NA	ID3 A
使う (tsukau)	use, spend, employ	80	85	93	91
飲む (nomu)	drink, take, eat, accept	42	90	98	93
行なう (okonau)	conduct, play, hold	33	94	88	88
応じる (oujiru)	answer, enter, meet	30	90	87	90
焼く (yaku)	burn, bake, roast, broil, cremate	27	93	89	93
解く (toku)	solve, undo, dispel	29	100	100	97
		Average Accuracy	92.0	92.5	92.0

ples. We gave, however, an easy way to "compile" the relevant background knowledge along with the ambiguous training examples into a modified set of training examples on which we were able to directly run ID3. Experiments comparing these approaches showed that the rules learned using the second approach with the ambiguity present in the training examples are almost as accurate as those obtained from ambiguity-free examples using Haussler's algorithm.

Overall, our experiments showed that using machine learning techniques yields rules that are highly accurate compared to the manually created rules. These results suggest that exploiting the reported inductive learning techniques will significantly accelerate the construction process of ALT-J/E's translation rules. Currently, the reported learning approaches are being included in a semi-automatic knowledge acquisition tool to be used in the actual development of the ALT-J/E system.

Acknowledgement: We wish to thank Dr. S. Ikehara for his continuous encouragement. This work was done while the first author was spending a post-doctoral year at NTT. He also thanks King Fahd University of Petroleum and Minerals, Saudi Arabia, for their support.

References

[Buchanan & Wilkins 1993] Buchanan, B. G. and Wilkins, D. C. (Eds.), *Readings in knowledge acquisition and learning*, Morgan Kaufmann, 1993.

[Haussler 1988] Haussler, D., "Quantifying inductive bias: AI learning algorithms and Valiant's learning framework", *Artificial Intelligence*, 26(2), 177-221, 1988.

[Ikehara et al. 1989] Ikehara, S., Miyazaki, M., Shirai, S. and Yokoo, A., "An Approach to Machine Translation Method based on Constructive Process Theory", *Review of ECL*, vol.37, No.1, 39-44, 1989.

[Ikehara et al. 1990] Ikehara, S., Shirai, S., Yokoo, A. and Nakaiwa, H., "Toward an MT System without Pre-Editing - Effects of New Methods in ALT-J/E", *Proc. of MT Summit-3*, 1990.

[Ikehara et al. 1991] Ikehara, S., Yokoo, A. and Miyazaki, M., "Semantic Analysis Dictionaries for Machine Translation", (in Japanese), *IEICE Report, NLC 91-19, Institute of Electronics, Information and Communication Engineers, Japan*, 1991.

[Kim & Moldovan 1993] Kim, J. T. and Moldovan, D. I., "Acquisition of Semantic Patterns for Information Extraction from Corpora", *Proc.of CAIA-93*, 171-176, 1993.

[Porter et al. 1990] Porter, B. W., Bareiss, R., and Holte, R. C. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45(3):229-263, 1990.

[Quinlan 1986] Quinlan, J. R. "Induction of Decision Trees", *Machine Learning*, 1(1):81-106, 1986.

[Sato 1991a] Sato, S., "MBT1: Example-Based Word Selection" (in Japanese), *Journal of Japanese Society for Artificial Intelligence*, vol.6, No.4, 1991.

[Sato 1991b] Sato, S., "MBT2: A method for Combining Fragments of Examples in Example-Based Translation" (in Japanese), *Journal of Japanese Society for Artificial Intelligence*, vol.6, No.6, 1991.

[Utsuro et al. 1992] Utsuro, T., Matsumoto, Y. and Nagao, M. "Lexical Knowledge Acquisition from Bilingual Corpora", *Proc. of the 14th International Conference on Computational Linguistics*, 581-587, 1992.

[Wilkins 1990] Wilkins, D. C. Knowledge base refinement as improving an incomplete and incorrect domain theory. In Kodratoff, Y. and Michalski, R. S., (eds), *Machine Learning: An Artificial Intelligence Approach*, vol. III, pp 493-514, Morgan-Kaufmann Publishers, 1990.