Acknowledgement

We wish to thank Dr. Shigeo Kaneda, Dr. Satoru Ikehara and Dr. Tsukasa Kawaoka for their continuous encouragement of this research. Masahiko Haruno and the anonymous referees provided helpful comments on an earlier version of this paper. The research reported here was supported in part by NTT, NSF grant IRI-9310413, ARPA grant F49620-92-J-0430, and AFOSR AASERT grant F49620-93-1-0569.

References

[Aldenderfer & Blashfield 1984] Aldenderfer, M., and Blashfield, R. 1984. Cluster Analysis. *SAGE University Papers.*

[Almuallim et al. 1993] Almuallim, H., Yamazaki T., AKiba, Y., Yokoo, A., Kaneda, S., and Kawaoka, T. 1993. Acquisition of Machine Translation Rules Using Inductive Learning Techniques. *IJCAI-93 Workshop Machine Learning and Knowledge Acquisition.*

[Almuallim et al. 1994] Almuallim, H., Akiba, Y., Yamazaki T., Yokoo, A., and Kaneda, S. 1994. A tool for the Acquisition of Japanese-English Machine Translation Rules Using Inductive Learning Techniques. *The 10th Conference of Artificial Intelligence for Applications.*

[Brown et al. 1992] Brown, P.F., Della Pietra, P.V., Lai, J.C., and Mercer, R.L. 1992 Class-Based n-gram Models of Natural Language, *Computational Linguistics,* Vol. 18, No.4, 467–479.

[Fisher 1987] Fisher, D. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning* 2:139-172.

[Grefenstette 1992] Grenfenstette, G., 1992 SEXTANT: Exploring unexplored contexts for semantic extraction from syntactic analysis, *Proceedings of the 30th Annual Meeting of ACL*, 324–326.

[Hindle 1990] Hindle, D. 1990 Noun classification from predicate argument structures, *Proceedings of the 28th Annual Meeting of ACL*, 268–275.

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

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

[Keene 1991] Donald Keene. 1991. Japanese-English Sentence Equivalents (Revised Version). Asahi Press.

[Michalski et al. 1986] Michalski, R., and Stepp, R. 1986 Conceptual clustering of structured objects: A goal-oriented approach. *Artificial Intelligence* 28:43-69.

[Miller et al. 1990] Miller, G., Beckwith, R., Felbaum, C., Gross, D., and Miller, K. 1990. Five Papers on Wordnet. *CSL Report 43*, Cognitive Systems Laboratory, Princeton University.

[Pagallo & Haussler 1990] Pagallo, G. & Haussler, D. 1990. Boolean feature discovery in empirical learning. *Machine Learning*, 5:71-100.

[Pazzani & Kibler 1992] Pazzani, M., and Kibler, D. 1992. The utility of knowledge in inductive learning. *Machine Learning*, 9(1):57-94.

[Pazzani et al. 1995] Pazzani, M., Yamazaki, T., and Merz, C. 1995. Acquiring and Updating Hierarchical Knowledge: Constructive Induction in Language Processing. (Unpublished).

[Quinlan 1990] Quinlan, J. R. 1990. Learning logical definitions from relations. *Machine Learning,* 5(3).

With the real data, the hierarchy created by our learning method from scratch resulted in rules that were more accurate than those learned using the handcrafted hierarchy. This result is marginally significant (t(29) = 1.7, p < .1). In addition, rules learned by updating the handcrafted hierarchy were on average 87.2% accurate, significantly more accurate than the accuracy (84.9%) of rules with the original hierarchy (t(29) = 3.796, p < .001). Moreover, this accuracy is significantly higher than the accuracy of rules created by our method learning from scratch (t(29) = 1.861, p < .005). Although these results are promising, the rules learned from real, ambiguous data are not as accurate as those obtained from the artificial data. Note that there are fewer examples of each English verb in the real data and the real data has additional complications because it contains potentially ambiguity. We are going to continue collecting real data from other sources such as newspapers to improve the accuracy of rules learned from real data.

7 Limitation

The hierarchy generated by our method experimentally showed similar ability to the original handcrafted hierarchy, however, some problems remain left unsolved.

One important limitation of the current approach is that the categories newly created in the hierarchy do not have meaningful names and the rules learned using such terms are not easy for a person to comprehend. A person has to explicitly give a label to each node so that it is understandable to another human.

Another limitation is that all necessary leaf nodes have to be prepared before learning. Other clustering techniques (e.g., [Brown et al. 1992]; [Hindle 1990]; [Grefenstette 1992]) might be useful to find a meaningful group from a set of noun words and regard it as one of leaf nodes.

Moreover, we have not addressed pruning of learned translation rules. With no pruning, too many specific rules may be learned. As a result, the hierarchy that is generated from these rules might contain redundant or incorrect nodes. In contrast, if the rules are pruned too much, fewer categories appear together in rules and it is not possible to create a hierarchy that contains enough nodes to make the needed distinctions when learning translation rules.

8 Related Work

The general topic of clustering has been studied extensively in the artificial intelligence field(e.g., [Fisher 1987]; [Michalski et al. 1986]). However, this work is not directly applicable to our problem. AI clustering methods work by computing some measure between features vectors that describe examples. In our case, examples are not described by such feature vectors but by similarity as represented by mutual information gain.

In natural language processing some approaches have been made to finding similarities among nouns. Hindle [Hindle 1990] uses mutual information between a verb and its object as a measure of degree of the association between the verb and an object. A measure of similarity among two nouns with respect to a single verb is defined to be the minimum noun similarity to the verb, and the overall similarity among two nouns is defined as the sum of similarities over all verbs. Grenfenstette [Grefenstette 1992] proposed a method of making noun clusters appropriate for each target domain by exploiting the kinds of nouns that appear as noun modifiers in the target domain. Both are intended to identify similar nouns, but they do not use similarity criteria as a metric for forming a hierarchy.

In constructive induction, a number of systems have analyzed the results of learning, and looked for opportunities to add new terms to simplify learned concepts (e.g Fringe [Pagallo & Haussler 1990]). We analyze internal disjunctions of learned translation rules in an attempt to identify general categories that represent commonly occurring disjunctions.

9 Conclusion

This paper reported our work on the acquisition and refinement of a semantic hierarchy through the use of clustering techniques. We extended the function of FOCL, a tool for learning translation rules, to handle the ambiguity of real data. We have shown that the constructive induction of terms representing general categories provides a useful bias for learning translation rules. Furthermore, the results on real and artificial data demonstrate that learning a hierarchy or updating an existing hierarchy can improve the generalization accuracy of systems that learn translation rules.

The main purpose of our method is to produce the semantic hierarchy appropriate for learning translation rules and to determine the method of constructing intermediate nodes given all leaf nodes. Currently various kinds of thesaurus such as ROGET and WordNet are available. It is true that they might be used for all kinds of purposes but they are not always proper for specific purposes such as learning translation rules. In practice, it is reasonable to modify available thesaurus in order to make them suitable for each domain. For example, in the text classification domain, intermediate nodes in a thesaurus might be useful for representing classification rules. In this case, a thesaurus for this domain can be constructed by a method similar to ours. Our proposed method suggests one direction for generating thesauri for various uses.

Table 2: Accuracy Result : 5 methods : For artificial Data

	Accuracy				
Japanese Verb	NO Hierarchy	Handcrafted Hierarchy	Learned Hierarchy	Incomplete Hierarchy	Update Hierarchy
Yaku	.827	.943	.897	.867	.937
Nomu	1.000	1.000	1.000	1.000	1.000
Okonau	.994	.997	.994	.997	.991
Oujiru	1.000	1.000	1.000	1.000	1.000
Toku	.999	.999	.999	.999	.999
Tsukau	.751	.938	.841	.848	.914
Ataru	.647	.913	.710	.718	.906
Atsumeru	.698	.997	.926	.938	.997
Eru	.996	.992	.992	.996	.996
Karamu	.675	1.000	.859	.897	.998
Kotaeru	.927	.999	.989	.984	.996
Nobiru	.853	1.000	.993	.972	1.000
Noru	.823	.907	.860	.89	.90
Nuru	.998	1.000	.998	1.000	1.000
Osamaru	.867	.999	.880	.995	.999
Oshimu	.738	.968	.797	.911	.963
Sasageru	.831	.997	.931	.830	.997
Soeru	.969	.987	.950	.985	.981
Taku	1.000	1.000	1.000	1.000	1.000
Taosu	.728	.953	.798	.917	.947
Tataku	.893	.875	.872	.89	.866
Umu	.859	.987	.923	.903	.987
Yaburu	.82	.987	.876	.88	.983
Yashinau	.713	.945	.732	.927	.946
Average	.859	.974	.906	.935	.971

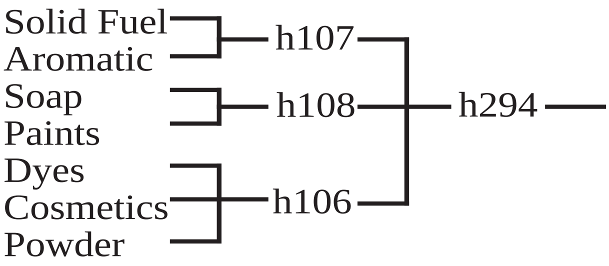

Figure 5: Part of the Generated Semantic Hierarchy

all intermediate nodes that appeared in rules learned with the handcrafted hierarchy were deleted. We call this the “Incomplete Hierarchy”.

Table 2 contains the results of this experiment. Rules learned with the incomplete hierarchy are less accurate than those learned with the handcrafted hierarchy, showing that deletions affect the learning of translation rules. Rules learned with the updated incomplete hierarchy were on average 97.1% accurate, which is significantly more accurate than the rules learned with the incomplete hierarchy (t(23) = 3.672, p < .005). Moreover, this accuracy is significantly higher than the accuracy with the hierarchy learned from scratch. (t(23) = 4.455, p < .001). This result shows that the method of updating an existing hierarchy yields a more useful hierarchy than learning a hierarchy from scratch.

Table 3: Accuracy Result : 4 methods : 30 J-Verbs : Real Data

Accuracy			
NO Hierarchy	Handcrafted Hierarchy	Learned Hierarchy	Update Hierarchy
.839	.849	.864	.872

6.3 Real Data

The accuracy was estimated by 10 fold cross validation as in the case of artificial data. Table 3 contains the result for real data. Note that for the real data, we updated the actual hierarchy used by ALT-J/E; no errors were introduced.

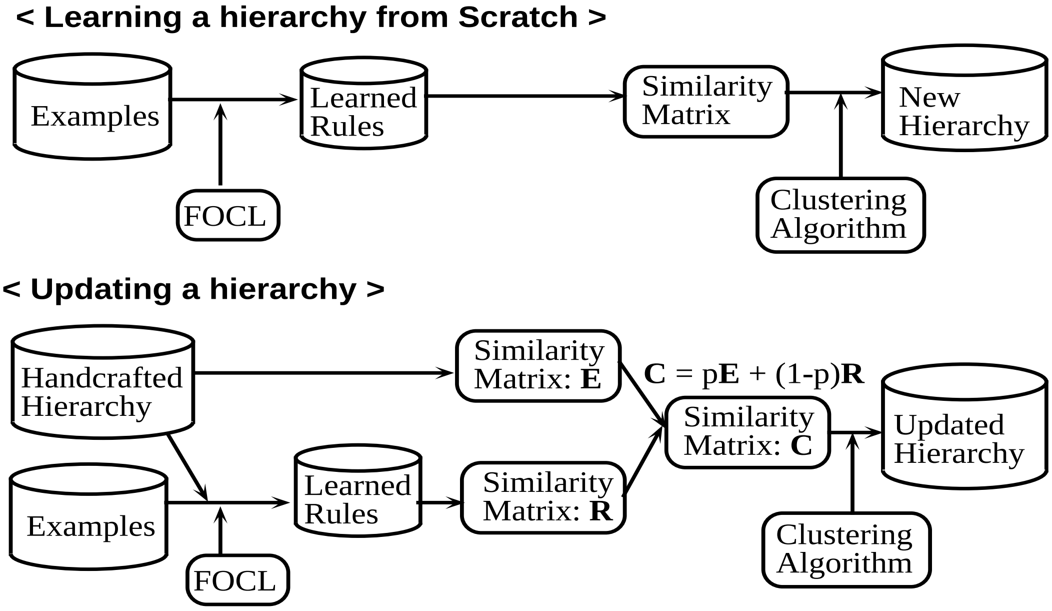

Figure 3: An overview of our proposed methods

the real training data since it was prepared by using a Japanese parser and no parser is 100% accurate.

6.2 Artificial data

6.2.1 Learning a hierarchy from scratch

The accuracy was estimated by 10 fold cross validation.[5] Table 2 shows the accuracies of rules generated for the artificial data under three conditions: "No Hierarchy", the "Learned Hierarchy" and the "Handcrafted Hierarchy". On average, with no hierarchy, the translations rules are 85.9% accurate, while with the handcrafted hierarchy the rules are 97.4% accurate ($t(23) = 5.26$, $p < .0001$). Moreover, the average accuracy with the learned hierarchy (90.6%) is substantially higher than learning with no hierarchy ($t(23) = 4.13$, $p < .001$). This result indicates the effectiveness of using the handcrafted hierarchy and the learned hierarchy for learning translation rules.

Figure 5 shows a part of the generated semantic hierarchy (where "Hxxx" indicates new nodes). The total number of generated nodes was 415. This hierarchy reproduces a part of the hierarchy created manually. This suggests that semantically reasonable nodes are created by our method. The rules learned with the generated semantic hierarchy are expressed with new nodes. Figure 4 shows a part of the learned rules for the Japanese verbs "Nomu" and "Taku". You can find the new nodes shown in Figure 5 in this rule.

```
nomu(Subject, Object, Everb) :-
one-isa(Object, {h294}), Everb = take.
nomu(Subject, Object, Everb) :-
one-isa(Object, {narration, lecture}, Everb
   = accept.
......
nomu(Subject, Object, Everb) :- Everb =
   drink.

taku(Subject, Object, Everb) :-
one-isa(Object, {h107, liquid fuel, gaseous
   fuel}),
Everb = burn.
....
```

Figure 4: Part of the rules learned for the Japanese verbs "Nomu" and "Taku"

6.2.2 Updating a hierarchy

To evaluate the procedure for updating a hierarchy with the artificial data, we purposely introduced errors into the handcrafted hierarchy by deleting some of the categories in the current hierarchy. In this experiment

[5] First 10 % of the examples for each verb were set aside for testing. Next we ran FOCL on the remaining nine tenths of the data, learning translation rules for the Japanese verbs without a hierarchy. Then hierarchy was created based on our proposed method. We then ran FOCL again on the same training data with the learned hierarchy. Finally, we tested the accuracy of the rules learned with this learned hierarchy on the one-tenth of the examples reserved for testing. This process was repeated ten times, each time using a different set of test examples.

1. Search the dissimilarity matrix to find the pair of categories that has maximum dissimilarity. Find all other categories that have the same maximum value in that column.
2. Combine the categories found in Step 1 to create a new node in the hierarchy.
3. The similarity between this node and the remaining categories is calculated by computing the average of the similarity between each of the categories used to form the new node and all others.
4. Update the matrix by adding the new node, deleting the categories used for creating the node, and assigning the similarity values computed by the previous step, to the new node.
5. If the matrix is empty, stop. Otherwise, go to step 1.

Table 1: The average-linkage clustering method

Next compare the rules learned by using this hierarchy with those learned without it. Suppose that only the categories C1 and C2 appear in the training examples. The form of rules learned without the hierarchy might be C1 or C2, while the form learned with the hierarchy might include generalized node H1. Therefore, when a test example includes C3, the example will be properly classified only if the rules were learned using the semantic hierarchy.

5 Updating an existing hierarchy

The goal here is to take advantage of human supplied knowledge when it proves useful in guiding the translation task, while tolerating omissions or errors in the hierarchy. If a hierarchy has already been developed manually, it is reasonable to utilize the human knowledge invested in the hierarchy by refining it rather than recreating it from scratch.

The hierarchy updating process will take as input an existing hierarchy, and the set of translation rules learned using that hierarchy. This procedure is as follows.

First, the existing hierarchy is converted into a matrix that represents the similarity among the leaves of the hierarchy. There are an infinite number of matrices that could be constructed from a hierarchy, such that when the average-linkage clustering method were run on that matrix the original hierarchy would be created. We created one such matrix in which the similarity between two leaves is represented by the minimum number of is-a links that need to be traversed to reach the first common ancestor of the two nodes. We will call the matrix E.

Second, general nodes in learned rules are replaced by a disjunction of their children since rules learned with an existing hierarchy may contain general nodes. A similarity matrix is then generated by computing the mutual information between specific nodes used in internal disjunctions in the same way as mentioned in the previous section. This matrix is called R.

Third, these two matrices are combined. To combine these matrices, first each entry in the matrix is normalized to the range [0,1]. The combined similarity matrix C is obtained by the weighted addition of these two matrices: $C_{ij} = pE_{ij} + (1-p)R_{ij}$
where p is a parameter that represents how much weight should be placed upon the existing hierarchy. In our experiments, we set p to 0.5 so that both sources of information have equal weight. We have obtained good results with this value and have not experimented with other possible values.

Finally, a new hierarchy is formed from this combined matrix by the average-linkage method. An overview of our approach is shown in Figure 3.

6 Experimental results

6.1 Experimental setting

In this experiment, we used two kinds of data set. One used artificially created data.[4] We generated 300 examples for each English verb's rule. We dealt with 24 different Japanese verbs corresponding to about 100 English verbs. A total of approximately 30,000 examples were generated.

The other set consisted of natural data. We collected Japanese-English corpora from example sentence translations in bilingual dictionaries (e.g. [Keene 1991] etc.), processed these corpora with the parser used in ALT-J/E, and then generated learning data. The total number of collected corpora amounted to approximately 48,000. It included more than 5,000 kinds of Japanese verbs. The Japanese verbs which include more than 30 examples were chosen from among them in this experiment. 30 Japanese verbs were selected corresponding to 120 English verbs. About 3400 examples were utilized in all.

The real data is more complex than the artificial data in a variety of ways. First, the real data typically has several possible values for each subject and object, while the artificial data has one. Second, it's not clear whether the existing hierarchy is adequate for the the real data. Finally, some noise may be introduced into

[4] This data set was generated by using the translation rules and the semantic hierarchy created by human experts that are currently used in the ALT-J/E system. Namely, first, it searches the hierarchy to find all leaf nodes of categories used in the rule for each attribute such as N_1 and N_2. Then, it chooses one leaf node among them randomly as a value for the attribute.

indicates that "tsukau" is translated to "spend" when one sense of the subject is a kind of "agents" and one sense of the object is "time" or "money."

Here we concentrate on describing FOCL's inductive learning component based on FOIL [Quinlan 1990]. FOCL inductively learns classification rules by constructing a set of Horn Clauses in terms of a set of background predicates (such as one-isa). It creates clauses in a set-covering manner until each positive example is covered by at least one clause. A clause is learned using hill-climbing search by adding the literal yielding the maximum information gain to the clause body.

3.2.2 An efficient way of handling hierarchical knowledge

Hierarchical knowledge is efficiently handled as background knowledge by exploiting the hierarchical relationships among the categories. We extended FOCL to compute the information-gain of literals of the form "one-isa(Role, Categories)" efficiently. This was accomplished by associating counters with each node in the hierarchy where each node describes the number of positive and negative training examples. The training data is traversed once for each variable (e.g., Subject and Object), to set counter values on the leaves of the hierarchy for each possible sense, and these values are propagated up the hierarchy. Care must be taken to insure that a parent node is incremented only once if a word has two senses corresponding to different children of the node. The information gain of only those nodes that satisfy at least one positive example is computed by using the counts stored at that node.

3.2.3 Creation of an internal disjunction based on hill-climbing

FOCL was modified to create the internal disjunctions in a greedy, set covering manner. First, the literal with the maximum information gain of the form one-isa(V, c_1) is found. Next, all literals of the form one-isa(V, $\{c_1, c_i\}$) are checked where c_i satisfies at least one positive example, and neither one-isa(c_1, c_i) nor one-isa(c_i, c_1) is true. If the information gain of one of these literals is not greater than that of one-isa(V, c_1), then one-isa(V, c_1) is returned as the literal with the maximum gain. Otherwise, the process of adding constants to the internal disjunction continues until adding another constant to the set does not increase the information gain. Using literals of the form one-isa(V, $\{c_1, ..., c_n\}$) is useful for representing node disjunction in the existing hierarchy (e.g., one-isa(Object, {fish, seafood}). It is also useful when there is no hierarchy. In this case, one-isa represents disjunctions of specific constants (e.g., one-isa(Object, {salmon, swordfish, lobster, shrimp}).

4 Learning a Semantic Hierarchy from scratch

As mentioned above, a semantic hierarchy plays an important role from the view of offering the more general terms needed for generating an appropriate level of generalization for translation rules. It is desirable that a semantic hierarchy has general nodes that include a proper group of leaf nodes as the most specific categories. Thus we regard the task of learning a semantic hierarchy as generating the appropriate nodes when leaf nodes are given.

We describe how to create a semantic hierarchy by using the translation rules learned by FOCL. After learning rules without a hierarchy by FOCL, the frequency with which two terms occur together in an internal disjunction of these rules (i.e., in a literal of the form one-isa(V, $\{c_1, ..., c_n\}$) is used as a measure of the similarity between the terms. In particular, the mutual information gain ratio is used as follows. Let $p(c_i)$ be the probability that the term c_i appears in an internal disjunction of all learned translation rules. This is estimated by dividing the number of times that c_i appears in an internal disjunction by the total number of internal disjunctions. The probability that both c_i and c_j appear in an internal disjunction ($p(c_i \& c_j)$) is estimated in the same manner. The mutual information gain measure $log(p(c_i \& c_j)/p(c_i)p(c_j))$ is used as the measure of the similarity between terms c_i and c_j.

A triangular matrix is created to store the value of the mutual information for all pairs.[3] A standard statistical clustering algorithm, the average-linkage method [Aldenderfer & Blashfield 1984] is then used to create a hierarchy. This method is normally used to create binary hierarchies. We modified the method slightly to create hierarchies in which a node may have any number of children, provided all pairs of the children have the exact same value of the mutual information gain measure. The clustering algorithm is described in Table 1.

The generated semantic hierarchy works well to improve the accuracy of the learned rules in the following case:

Suppose that the categories C1, C2, and C3 appear together in several rules learned without a hierarchy and are used to generate a hierarchy. In this case, this proposed method produces new node H1, which consists of C1, C2, and C3.

[3] In the statistical literature, such a matrix is called a dissimilarity matrix, since a value of 0 indicates that two terms are unrelated, and large values indicate a strong relationship between the two terms. A similarity matrix is one in which 0 indicates that the two items are identical and larger values indicate a difference between the two items. A similarity matrix might be formed by taking the Euclidean distance between two feature vectors.

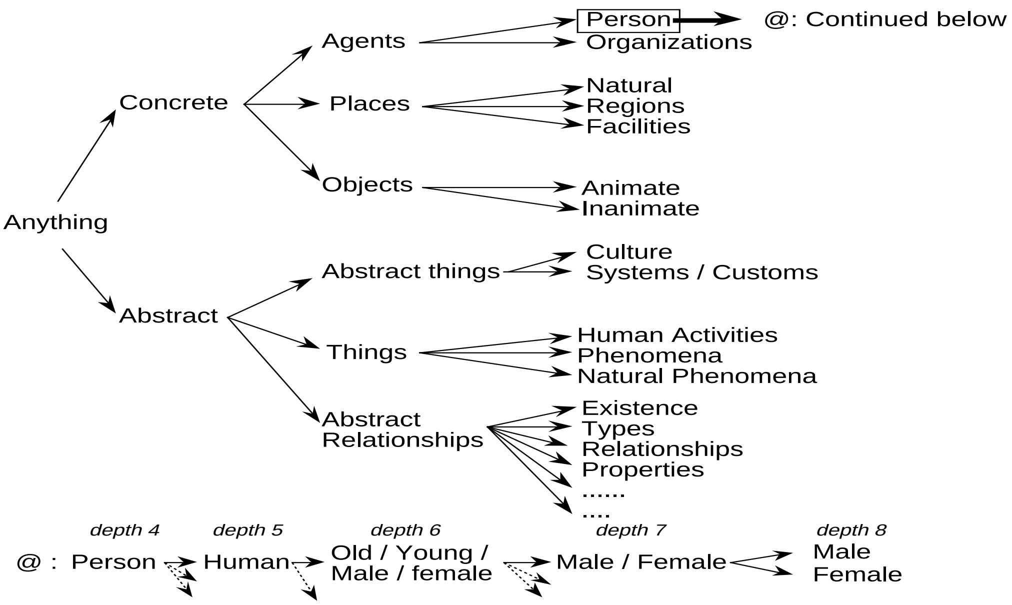

Figure 1: The upper levels of the Semantic Hierarchy in ALT-J/E

IF			THEN			
	J-Verb	= "TSUKAU"		Subj	$= N_1$
	N_1 (Subj)	≡ "Agents"		E-Verb	="spend"
	N_2 (Obj)	≡ "Time" or "Money"		Obj	$= N_2$
IF			THEN			
	J-Verb	= "TSUKAU"		Subj	$= N_1$
	N_1 (Subj)	≡ "Agents"		E-Verb	="employ"
	N_2 (Obj)	≡ "People"		Obj	$= N_2$
IF			THEN			
	J-Verb	= "TSUKAU"		Subj	$= N_1$
	N_1 (Subj)	≡ "Agents" or "Artificial Objects"		E-Verb	="use"
				Obj	$= N_2$
	N_2 (Obj)	≡ "Anything"			

Figure 2: Translation rules for the Japanese verb "TSUKAU". These rules were composed manually by human experts. Symbol " ≡ " indicates "an instance of".

3.2 An extension to FOCL

3.2.1 Representation in FOCL

We use FOCL [Pazzani & Kibler 1992] to learn translation rules from examples. In related work [Pazzani et al. 1995], we considered the simpler case of each attribute N_i such as the subject and object having only a single value. Here, we consider the more complex case where N_i has more than one value. In our earlier work, the predicate "isa(Term, C)" was introduced to indicate that the variable Term is an instance of the category C, such as "isa(dollar, money)". In this paper, when learning from potentially ambiguous data, we use a predicate "one-isa(TermSet, C)" to represent the fact that one of the terms in TermSet is an instance of category C, such as one-isa({asset, metal, medal}, money). This predicate is very useful when dealing with real world data in which words may have more than one possible meaning. Moreover, we have extended the semantics of the "one-isa" predicate so that the second argument may be a set of categories. In this case, one-isa(TermSet, CategorySet) is true if one term in TermSet is an instance of one of the categories in CategorySet. This extension enables the "one-isa" predicate to allow a form of internal disjunction. A translation rule can be represented as follows.

tsukau(Subject, Object, Everb) :-
one-isa(Subject, {agents}),
one-isa(Object, {time, money}), Everb =
spend.

utilized to guide the construction of the new hierarchy. This method uses some of the procedures used in the method of learning a hierarchy from scratch.

In the remainder of this paper, we give a brief description of the machine translation task. We then show how to represent translation rules in FOCL and how to extend FOCL to handle hierarchical knowledge efficiently. We propose a method for creating a semantic hierarchy from scratch. Next, we extend the method for creating a semantic hierarchy to update an existing incomplete hierarchy. Finally, we show experimental results and related works.

2 Background

We describe the machine translation task briefly by examining an experimental Japanese-English translation system, ALT-J/E [Ikehara et al. 1989, Ikehara et al. 1990], as an example. In this work, we are concerned with the following ALT-J/E components: the Semantic Hierarchy, Noun Dictionary, and the Translation Rules.

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. The current version of ALT-J/E's Semantic Hierarchy is 12 levels deep and has 2715 nodes, which consist of 790 intermediate nodes and 1925 leaf nodes.

The **Noun Dictionary** maps each Japanese noun to its appropriate semantic category in the hierarchy. For example, the Noun Dictionary states that the noun "maguro", which means "tuna" in English, is an instance of the category "fish" and also an instance of the category "food".

The **Translation Rules** in ALT-J/E associate Japanese patterns with English patterns. Currently, ALT-J/E holds roughly 10,000 of these rules. 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 "TSUKAU", the category of its subject is "Agents", and the category of its object is "Time" or "Money", then the following English pattern is to be used:

Subject "spend" Object.

Note that in this case the Japanese verb "TSUKAU" is translated into the English verb "spend". This same Japanese verb can also be translated into the English verbs "employ" or "use", depending on the context. These cases are handled by the two other rules given in Figure 2.

3 Learning Translation Rules with FOCL

3.1 The Learning Task

In this section we describe the task of learning translation rules[1] from examples. Each learning session concentrates on one Japanese verb, where it is desired to construct rules for mapping the verb in a given context to the appropriate English verb.

As an example, consider the Japanese verb "Tsukau" in the sentence "Shachou ga kane wo Tsukau", which means "The president spends money". The following pair is given after parsing (which is carried out by ALT-J/E's parser):

$$\langle\ [\ \ \textbf{J-V}\textsc{erb} = \text{Tsukau}\ ,\ \textsc{Subject} = \text{Shachou}, \\ \textsc{Object} = \text{Kane}\ ],\ \textbf{E-V}\textsc{erb} = \text{spend}\ \rangle.$$

It is usually the case that a given Japanese noun has more than one possible meaning. By looking in the Semantic Dictionary of ALT-J/E, the possible semantic categories for "Shachou" are "Manager"and "Chief", and those for "Kane" are "Asset", "Metal", and "Medal". Thus, this example is finally given to the learning algorithm in the following form to create rules for translating "tsukau:"

$$\langle\quad [\quad \textsc{Subject} \equiv \{\ \text{Manager, Chief}\ \}, \\ \textsc{Object}\ \equiv \{\ \text{Asset, Metal, Medal}\ \}]\ , \\ \textbf{E-V}\textsc{erb} =\ \text{spend}\qquad \rangle,$$

where $N \equiv S$ indicates that sentence component N is an instance of each category $s \in S$. As you can see in the example shown above, this example includes ambiguity in the sense that each of the attributes **Subject** and **Object** is assigned a set of possible values rather than a single value. The general format of the training examples is as follows:

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

where each N_i[2] represents a component of the sentence (subject, object, etc.), and a_i, b_i, and c_i are semantic categories.

The job of the learning algorithm is to find the rules such as those in Figure 2.

[1] This work is to learn what we call "verb selection rules". A verb selection 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 verb selection rule is that the latter has only an English verb rather than a full English pattern as its right-hand side.

[2] N_i is a variable which indicates a deep case frame. 14 kinds of variables including subject and object are defined in the current system.

Learning Hierarchies from Ambiguous Natural Language Data

Takefumi Yamazaki
NTT Communication Science Laboratories
1-2356 Take, Yokosuka-Shi
Kanagawa-Ken, Japan 238-03
yamazaki@nttkb.ntt.jp

Michael J. Pazzani and **Christopher Merz**
Department of Information and Computer Science,
University of California, Irvine, CA, 92717
{pazzani, cmerz}@ics.uci.edu

Abstract

This paper presents a method for acquiring a semantic hierarchy and updating an incomplete hierarchy. The creation of a comprehensive hierarchy is one important step in constructing a system for translating Japanese texts into English. The hierarchy is used to bias the learning of rules that indicate the English translation of a Japanese verb. The task is particularly challenging because training examples are ambiguous in the sense that each of the attributes forming an example may have a set of potential values. Each value corresponds to a different word meaning. When acquiring a hierarchy from scratch, translation rules are learned by an inductive learning algorithm in the first step. A new hierarchy is then generated by applying a clustering method to internal disjunctions of the learned rules and new rules are learned biased by this hierarchy. When updating an existing manually-constructed hierarchy, we take advantage of its node structure. We report experimental results showing that the semantic hierarchies generated by our method yield learned translation rules with higher average accuracy.

1 Introduction

Hierarchical knowledge is an important part of many artificial intelligence systems. In most natural language processing tasks, hierarchical knowledge is essential to capture general linguistic rules that apply in a variety of contexts. For example, the WordNet system [Miller et al. 1990] contains a lexical inheritance system of approximately 33,400 terms. Similarly, ALT-J/E, an experimental Japanese-English translation system developed at Nippon Telegraph and Telephone Corporation (NTT), uses a semantic hierarchy to aid in the process of translating texts. The research reported in this paper centers on the important task of automatically acquiring this hierarchical knowledge.

ALT-J/E contains a large collection of translation rules (currently over 10,000) and a large semantic hierarchy (about 3,000 nodes). Each of the translation rules associates one Japanese sentence pattern with an appropriate English pattern. Each rule is expressed in terms of "semantic categories" which are nodes of the semantic hierarchy. 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.

Creating new translation rules has proven to be extremely difficult and time-consuming because this task requires considering the huge space of possible combinations of nodes in the hierarchy. Previous research investigated the application of inductive learning techniques for the acquisition of translation rules [Almuallim et al. 1993, Almuallim et al. 1994]. Experimental results showed that this method was promising in facilitating the construction of rules. However, this method used the semantic hierarchy created by a human expert as it was and did not consider learning this hierarchy.

Learning the semantic hierarchy is essential and crucial in the machine translation task because the quality of learned translation rules is determined by the quality of this hierarchy and it is very difficult for a person to create and maintain a useful hierarchy.

In this paper, we propose two methods of learning a semantic hierarchy: learning a hierarchy from scratch and updating an existing hierarchy. In learning a hierarchy from scratch, translation rules are learned without using a hierarchy by applying extended FOCL [Pazzani & Kibler 1992] in the first step. Statistical clustering is then used to generate a hierarchy from the learned rules. The phase of hierarchy generation can be regarded as a kind of constructive induction in that general hierarchy terms serve as new predicates that can bias subsequent rule learning. In updating an existing hierarchy, its node structures are