Chapter 1

Introduction

This thesis extends a conventional framework of *Inductive Logic Programming (ILP)* [35] to that of hierarchically sorted Inductive Logic Programming and applies the extended ILP to information extraction tasks, which are a part of natural language processing. In Part I of this thesis, the ILP framework is extended to hierarchically sorted ILP, and Part II describes its application to information extraction.

ILP is a promising approach to *knowledge-level learning* from real-world examples such as a wide variety of facts contained in on-line newspapers or large-scale databases. The following advantages of ILP support this claim:

1. ILP learns rules on the basis of *background knowledge.*

2. Learned rules are represented in the form logic programs [32] that are comprehensible to humans.

3. Consistency between learned rules and existing knowledge can be checked with the properties of predicate calculus.

4. The properties of predicate calculus can be incorporated in learning algorithms in order to enhance learning speed and accuracy of learned rules.

ILP has been applied to a wide variety of fields, such as finite element mesh design [14], redesign of airplane panels [45], prediction of toxicology [61], and prediction of protein secondary structure [39].

However, major extensions are required for ILP systems to handle real-world relations found in daily life. First, classes (or *sorts*) of concepts and relations of the classes play a very important role in learning rules in daily life. Therefore, ILP systems should attach sort information to hypotheses in order to obtain comprehensible rules with an appropriate generality. Second, even if background knowledge includes thousands of hierarchically structured sorts, ILP systems must not deteriorate system performance.

While special treatment of hierarchical sort information is crucial to real-world applications, few research efforts have investigated the inductive learning of hypothesis clauses with sorts which are directly linked to sort hierarchy.

This thesis presents three approaches that deal with how to use sort information to learn hypotheses with sorts. The goal is to generate logic programs with sorts from concrete examples and background knowledge that has sort hierarchy.

Intuitively, "logic programs with sorts" means logic programs whose variables have sort information. An example will be more informative. Suppose you have positive examples and background knowledge of people who are happy. The goal of learning is to generate the following hypothesis:

$$happy(X : human) :- like(X, Y : object), get(X, Y).$$

This rule means that human X is happy if X gets object Y, which X likes. The variables in this hypothesis are given intermediate levels of generality between a constant and a free variable; X has the sort *human* and Y has the sort *object*. Restricting domains of variables with sorts structured by sort hierarchy can achieve an appropriate generality of hypotheses. We can also expect that sort restrictions to variables would greatly enhance readability and comprehensibility.

The following example illustrates that sort information is essential to learning when a human could learn rules of an appropriate generality from a small number of examples through sort hierarchy.

Example 1 (*Sample learning of "speak"*)

Suppose that a person knows the following relations :

- *Positive example:*
  { *speak(Jack,English)*, *speak(Jun,Japanese)* }

- *Background knowledge:*
  { *grew_in(Jack,UK)*, *grew_in(Jun,Japan)*,
  official_lang(UK,English), *official_lang(Japan,Japanese)* }.

Existing ILP systems might produce the following clause:

speak(X,Y) :− *grew_in(X,Z)*, *official_lang(Z,Y)*.

However, this clause must be over-general because it says that everything that grew up in a country, even a dog or a cat, speaks that language. Sorts effectively solve this problem. In this case, we expect the following relations with sorts to be produced:

speak(X:human, Y:language) :−
grew_in(X,Z:country), *official_lang(Z,Y)*,

where *Var:T* denotes that every variable *Var* appearing in a clause is the sort *T*. In other words, this clause represents "A human speaks a particular language if s/he grew up in a country whose official language is the language of the country".

To learn this relation, we have to give the following additional background knowledge to the learner:

{ *Jack*$\preceq$*male*, *Jun*$\preceq$*female*,*male*$\preceq$*human*,*female*$\preceq$*human*,
English$\preceq$*language*,*UK*$\preceq$*country*,*Japanese*$\preceq$*language*,
Japan$\preceq$*country* }

Note that $A \preceq B$ denotes A is a sub-sort of B, which is commonly represented as A is_a B in the AI community.

Adding *is_a* relations to background knowledge seems to lead us to our goal, but conventional ILP systems virtually fail to produce hypotheses with appropriate sorts even though background knowledge incorporates sorts and *is_a* relations. This is mainly because it is not realistic to handle a large number of fragments of a large-scale sort hierarchy as ordinary background knowledge. The sorts have a tree or lattice structure. Selecting appropriate sorts for variables requires traversing the structure of the sort hierarchy when generalizing and specializing a hypothesis. Therefore, how to use the structure of the sort hierarchy is a key to success for developing new learning methods that efficiently produce hypotheses with sorts.

If ILP can be successfully extended, it becomes applicable to natural language processing, particularly, information extraction. Information extraction involves extracting key information from a text corpus, such as newspaper articles or WWW pages. For instance, in news articles reporting a corporate merger, the merging companies' names, merger date, and merger reasons are key information to be extracted.

Domain dependence has been a serious problem for IE system developers. As an example, Umass/MUC-3 needed about 1500 person-hours of skilled labor to build the IE rules represented as a dictionary [29]. Worse, new rules have to be constructed from scratch when the target domain is changed. FASTUS needed three and a half weeks for constructing its domain-dependent part [5].

To cope with this problem, some pioneers have been studying methods of learning information extraction rules for years. Along these lines, the extended ILP systems will be applied to the learning of IE rules by which key information is extracted from news articles. Hierarchically sorted ILP best matches the learning of IE rules because taxonomy (*i.e.*, sort hierarchy) can be a guide in controlling generality of IE rules.

On the other hand, in the field of information extraction, applying ILP leads to a new direction in the learning of IE rules. This is the learning of semantic-level IE rules rather than superficial syntactic IE rules. From the viewpoint of engineering, it is meaningful that we can automatically generate IE rules from only several manually created examples.

With this motivation, this thesis describes methods to extend the ILP framework to deal with hierarchical sorts and an application of the extended ILP to information extraction.