Part II

Application to Information Extraction

Chapter 8

Applying Extended Inductive Logic Programming to Information Extraction

8.1 Overview

This section presents a brief introduction of our target application: information extraction (IE). The task of information extraction involves extracting key information from a text corpus, such as newspaper articles or WWW pages, in order to fill empty slots of given *templates*. Information extraction techniques have been investigated by many researchers and institutions in a series of Message Understanding Conferences (MUC), which are not only technical meetings but also IE system contests on information extraction, conducted on common benchmarks.

The input of the information extraction task is a set of natural language texts (usually newspaper articles) with an empty *template*. In most cases, the articles describe a certain topic, such as corporate mergers or terrorist attacks in South America. The given templates have some slots which have field names, *e.g.*, "company name" and "merger date". The output of the IE task is a set of filled templates. IE tasks are highly domain dependent because the rules and dictionaries used to fill values in the template slots depend on the domain.

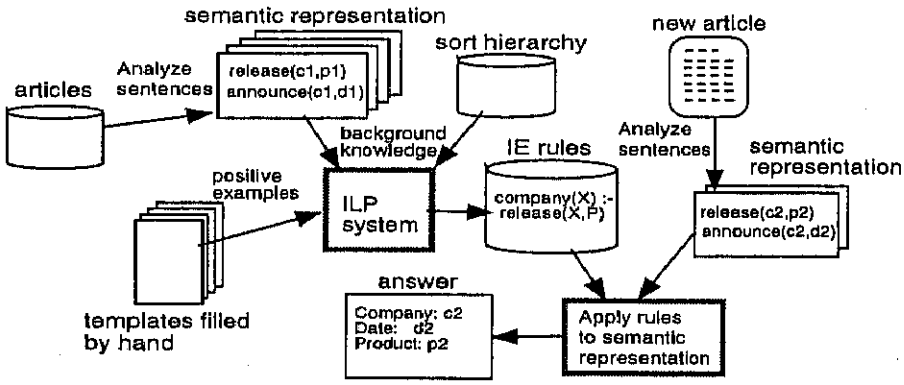

Figure 8.1: Block Diagram of IE using ILP

8.2 Problems in IE

The domain dependency 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 domain dependent part [5].

To cope with this problem, some pioneers have studied methods to learn information extraction rules. On this background, we selected the IE task for an application of extended ILP. An IE task is appropriate for our application because natural languages contain a vast variety of nouns relating to a taxonomy (*i.e.*, sort hierarchy).

8.3 Our Approach to IE Tasks

This section describes our approach to IE tasks. Figure 8.1 is an overview of our approach to learning IE rules using an ILP system from semantic representation. First, training articles are analyzed and converted into *semantic representation*, which are filled *case frames* represented as atomic formulae. Training templates are prepared by

hand as well. The ILP system learns IE rules in the form of logic programs with sort information. To extract key information from a new article, semantic representation automatically generated from the article is matched by the IE rules. Extracted information is filled into the template slots.

8.4 Evaluation Metrics

Three kinds of evaluation metrics are commonly used in the IE community. Those are *recall*, *precision* and *F-measure*.

$$Precision = \frac{the\ number\ of\ correct\ system\ outputs}{the\ number\ of\ system\ outputs}$$

$$Recall = \frac{the\ number\ of\ correct\ system\ outputs}{the\ number\ of\ all\ correct\ answers}$$

Intuitively, precision is accuracy, which means what percentage of system outputs are correct. Recall is coverage ratio, which means what percentage of the correct answers of a problem is covered by system outputs. Precision and recall correspond to the *Type I error rate* and the *Type II error rate*, respectively, that are used in significance testing in statistics.

There is a tradeoff between recall and precision. The higher precision an implementer tries to achieve by tuning system parameters, the lower recall becomes, and vice versa.

To express system performance in a single measure, F-measure was invented. Given precision P and recall R, F-measure F is:

$$F = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R}.$$

Originally, van Rijsbergen [62] defined effectiveness measure E

$$E = 1 - \frac{1}{\alpha(\frac{1}{P}) + (1 - \alpha)\frac{1}{R}}.$$

By transforming it according to $\alpha = 1/(\beta^2 + 1)$ so that $\partial E/\partial R = \partial E/\partial P$ *when* $P/R = \beta$, E will be:

$$E = 1 - \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R}.$$

David Lewis [30] defined F-measure as $F = 1 - E$ so that larger numeric values are better[1] .

If $\beta < 1$, a user attaches more importance to recall. If $1 < \beta$, a user attaches more importance to precision. If $\beta = 1$, a user puts the same importance to recall and precision. Usually, β is set to 1 and

$$F = \frac{2 \times P \times R}{P + R}$$

is used as F-measure. This thesis follows the suit.

8.5 Natural Language Processing Resources

8.5.1 The Semantic Attribute System

The semantic attribute system of "Goi-Taikei — A Japanese Lexicon" [22, 26] is compiled by the NTT Communication Science Laboratories for the use of a Japanese-to-English machine translation system, ALT-J/E [21]. The semantic attribute system is a hierarchical concept thesaurus represented as a tree structure in which each node is called a *semantic category*. Each edge in the tree represents an *is_a* relation between two categories. The semantic attribute system is 12 levels deep and contains about 3,000 semantic category nodes. More than 300,000 Japanese words are linked to the category nodes.

Figure 8.2 shows the structure of our sort hierarchy [20]. The hierarchy is a sort of concept thesaurus represented as a tree structure in which each node is called a category (i.e., a sort). An edge in this structure represents an *is_a* relation among the categories. For example, "Agents" and "Person" are both categories. The edge between these two categories indicates that any instance of "Person" is also an instance of "Agents." The current version of the sort hierarchy is 12

[1] According to [31], his calling 1 - E by the symbol "F" was the result of a mistake that he misinterpreted other equation with symbol F in [62] as a definition of $1 - E$.

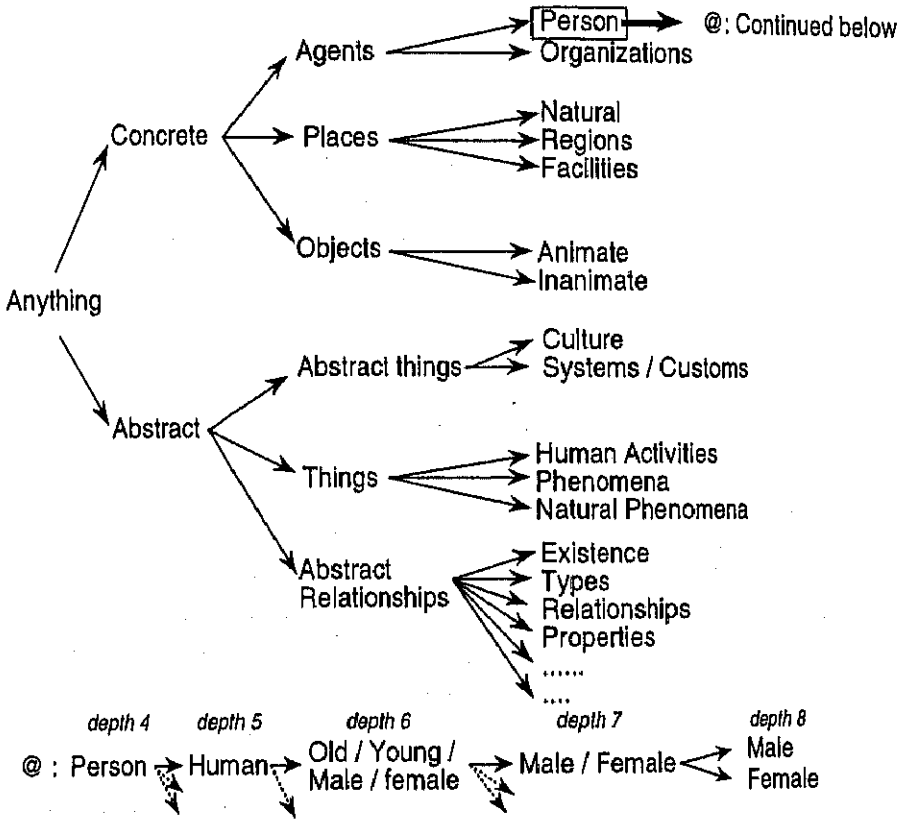

Figure 8.2: Upper Levels of a Sort Hierarchy

levels deep and contains about 3000 category nodes. This level of detail is necessary for a semantic analysis that enables real-world text understanding [20].

8.5.2 Verb Case Frame Dictionary

The Japanese-to-English valency pattern dictionary of "Goi-Taikei" [23, 26] was also developed for ALT-J/E. The valency dictionary contains about 15,000 case frames with semantic restrictions on their arguments for 6,000 Japanese verbs. Each case frame consists of one predicate and

one or more case elements that have a list of semantic categories.

8.5.3 Natural Language Processing Tools

The natural language processing components of ALT-J/E were used for the purpose of text analysis. These include the morphological analyzer, the syntactic analyzer, and the case analyzer for Japanese. The components are robust and generic tools, mainly targeted to newspaper articles.

Generic Case Analyzer

Let us see ALT-J/E's generic case analysis [57] in more detail. The case analyzer reads a set of parse tree candidates produced by the Japanese syntactic analyzer. The parse tree is represented as a dependency of phrases (*i.e.*, Japanese *bunsetsu*).

First, it divides the parse tree into unit sentences, where a unit sentence consists of one predicate and its noun and adverb dependent phrases. Second, it compares each unit sentence with a verb case frame dictionary. Each frame consists a predicate condition and several case elements conditions. The predicate condition specifies a verb that matches the frame, and each case-role has a case element condition which specifies particles and semantic categories of noun phrases. The preference value is defined as the summation of noun phrase preferences which are calculated from the distances between the categories of the input sentences and the categories written in the frames. The case analyzer then chooses the most preferable parse tree and the most preferable combination of case frames.

The valency dictionary also has case-roles (Table 8.2) for noun phrase conditions. The case-roles of adjuncts are determined by using the particles of adjuncts and the semantic categories of noun phrases.

As a result, the output of the case analysis is a set of case frames for each unit sentence. The noun phrases in frames are labeled by case-roles in Table 8.2.

For simplicity, we use case-role codes, such as N1 and N2, as the labels (or slot names) to represent case frames. The relation between sentences and case-roles is described in detail in [20].

Logical Form Translator

We developed a logical form translator FEP that generates semantic representation expressed as atomic formulae from the case frames and parse trees. For later use, document ID and tense information are also added to the case frames.

For example, the case frame in Table 8.1 is obtained after analyzing the following sentence of document D1:

"*Jakku*(Jack) *ha sutsukesu*(suitcase) *wo shokuba*(the office) *kara*(from) *kuko*(the airport) *ni*(to) *hakobu*(carry)"

("Jack carries a suitcase from the office to the airport.")

Table 8.1: Case Frame of the Sample Sentence

predicate:	*hakobu* (carry)
article:	D1
tense:	present
N1:	*Jakku* (Jack)
N2:	*sutsukesu* (suitcase)
N4:	*shokuba* (the office)
N5:	*kuko* (the airport)

Table 8.2: Case-Roles

Name	Code	Description
Subject	N1	the agent/experiencer of an event/situation (*e.g.*, *I* throw a ball.)
Object1	N2	the object of an event (*e.g.*, I throw *a ball.*)
Object2	N3	another object of an event (*e.g.*, I compare it *with them.*)
Loc-Source	N4	source location of a movement (*e.g.*, I start *from Japan.*)
Loc-Goal	N5	goal location of a movement (*e.g.*, I go *to Japan.*)
Purpose	N6	the purpose of an action (*e.g.*, I go *shopping.*)
Result	N7	the result of an event (*e.g.*, It results *in failure.*)
Locative	N8	the location of an event (*e.g.*, It occurs *at the station.*)
Comitative	N9	co-experiencer (*e.g.*, I share a room *with him.*)
Quotative	N10	quoted expression (*e.g.*, I say *that ....*)
Material	N11	material/ingredient (*e.g.*, I fill the glass *with water.*)
Cause	N12	the reason for an event (*e.g.*, It collapsed *from the weight.*)
Instrument	N13	a concrete instrument (*e.g.*, I speak *with a microphone.*)
Means	N14	an abstract instrument (*e.g.*, I speak *in Japanese.*)
Time-Position	TN1	the time of an event (*e.g.*, I go to bed *at 10:00.*)
Time-Source	TN2	the starting time of an event (*e.g.*, I work *from Monday.*)
Time-Goal	TN3	the end time of an event (*e.g.*, It continues *until Monday.*)
Amount	QUANT	quantity of something (*e.g.*, I spend *$10.*)