PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1992,
edited by ICOT. © ICOT, 1992

Analogical Generalization

Takenao OHKAWA† Toshiaki MORI‡ Noboru BABAGUCHI‡ Yoshikazu TEZUKA‡

†*Education Center for Information Processing, Osaka University*
‡*Dept. of Communication Eng., Faculty of Eng., Osaka University*
2-1, Yamadaoka, Suita, Osaka, 565 Japan
e-mail: ohkawa@oucom5.oucom.osaka-u.ac.jp

Abstract

Approaches to learning by examples have focused on generating general knowledge from a lot of examples. In this paper we describe a new learning method, called analogical generalization, which is capable of generating a new rule which specifies a given target concept from a single example and existing rules. Firstly we formulate analogical generalization based on the similarity between a given example and existing rules from the logical viewpoint. Secondly, we give a new procedure of inductive learning with analogical generalization, called ANGEL. The procedure consists of the following five steps: (1) extending a given example, (2) extracting atoms from the example and selecting a base rule out of the set of existing rules, (3) generalizing the extracted atoms by means of the selected rule as a guide. (4) replacing predicates, and (5) generating a rule. Through the experiment for the system for parsing English sentences, we have clarified that ANGEL is useful for acquiring rules on knowledge based systems.

1 Introduction

Machine learning has a great contribution to improving performance through automated knowledge acquisition and refinement, and so far, various types of machine learning paradigms have been considered. In particular, learning from examples, which can form general knowledge from specific cases given as input examples, has been well studied and a lot of concerned methods have been proposed[Mitchell 1977, Dietterich and Michalski 1983, Ohkawa *et al.* 1991].

Generally, in learning from examples, we have to give a lot of examples to the learner. Why are so many examples required? We think the reason for this is that the bias for restricting the generalization is relatively weak, because it is independent of the domain. However, when a human being acquires new knowledge, he would not always require a lot of examples. As the case may be, he can learn from one example. We think this is because he decides a strong bias for the generalization according to the domain, and generalizes the examples based on the bias. That is, in order to generalize a few examples appropriately, a strong bias which depends on the domain is indispensable.

It is necessary to consider how the strong bias should be provided. Let us recall the behavior of a human being again. When acquiring new knowledge, he often utilizes similar knowledge which is already known. In other words, the existence of similar knowledge may help for him to associate new knowledge. This process is called analogy. Analogy is considered promising to realize learning from a few examples. Since analogy will be regarded as one of the most effective way for restriction on generalization, modeling its process will make it possible to provide a domain dependent bias.

In this paper, we propose a new learning method, called ANGEL (ANalogical GEneraLization), which is capable of generating a new rule from a single example. In ANGEL, both the rules and the examples are represented as logical formulas. We introduce the notion of analogy[Winston 1980], namely, the similarity between the example and the existing rules as the bias for the generalization[Mori *et al.* 1991]. The similarity is determined by comparing the atoms of both the example and the existing rules. Based on the similarity, firstly, ANGEL extracts atoms from the example and selects a rule out of the existing rules; next, it generates a new rule by generalizing the extracted atoms by means of the selected rule as a guide.

The next section describes the definition of analogical generalization. In this section we consider analogical generalization from the logical viewpoint. Section 3 gives the procedure of ANGEL which is a method for learning based on analogical generalization. In this section, we also give consideration to the experimental result of learning by ANGEL. Finally in section 4, we clarify the originality of ANGEL through its comparison to other related works.

2 Analogical generalization

To represent knowledge, we use the form which conforms to first order predicate logic. Two kinds of forms, called a fact and a rule, are provided. A fact is represented as an atom, while a rule is represented as a Horn clause, which is expressed in the form of

$$\alpha \leftarrow \beta_1, \ldots, \beta_n,$$

where $\alpha, \beta_1, \ldots, \beta_n$ are atoms. Letting r be a rule $\alpha \leftarrow \beta_1, \ldots, \beta_n$, we denote the consequence of rule r, namely α, by $cons(r)$, and denote the premise of rule r, namely $\beta_1, \ldots, \beta_n$, by $prem(r)$.

The underlying notion of analogical generalization is that a new rule is generated by generalizing an input example, which consists of facts, based on the similarity between the example and the existing rules. Before formulating analogical generalization, we define the similarity between two atoms, and next formalize the similarity between two finite sets of atoms.

2.1 Similarity between two atoms

First, we define some basic notations. A substitution is a finite set of the pair v/t, where v is a variable, t is a term, and the variables are distinct. Let $\theta = \{v_1/t_1, \ldots, v_n/t_n\}$ be a substitution and e be an expression, which is either a literal or a conjunction or disjunction of literals. Then $e\theta$ is the expression obtained from e by replacing each occurrence of the variable v_i in e by the term t_i. If S is a finite set of expressions and θ is a substitution, $S\theta$ denotes the set $\{e\theta \mid e \in S\}$.

Let θ be a substitution and S be a finite set of atoms. If $S\theta$ is a singleton, S is unifiable by θ and we write *unifiable*(S).

Now, we give the following two functions, and define the similarity between atoms by means of these functions. Let R be a set of existing rules, and α and α' be atoms.

Definition 1 (R-deducible set)

$$\Phi(R,\alpha) \stackrel{\text{def}}{=} \{\beta \mid R \cup \{\alpha\} \vdash \beta, \beta \text{ is an atom}\}.$$

Definition 2 (R-similar set)

$$\Psi(R,\alpha,\alpha') \stackrel{\text{def}}{=} \{\beta \mid \beta \in \Phi(R,\alpha), {}^{\exists}\beta' \in \Phi(R,\alpha'), \textit{unifiable}(\{\beta,\beta'\})\}.$$

R-deducible set means all of newly obtained information when a certain fact has been known. Thus the intuitive meaning of R-similar set is newly obtained information in common when each of two distinct facts has been known. Therefore we can say that R-similar set represents the relevance between two facts under the background knowledge.

Definition 3 (Similarity between atoms) *Let* α, α_1 *and* α_2 *be atoms. If the following relation holds,* α **is more similar to α_2 than α_1 with respect to R.**

$$\Psi(R,\alpha,\alpha_1) \subset \Psi(R,\alpha,\alpha_2)$$

And if the following holds, **the similarity between α and α_1 is equal to the similarity between α and α_2 with respect to R.**

$$\Psi(R,\alpha,\alpha_1) = \Psi(R,\alpha,\alpha_2)$$

Since R-similar set reflects the relevance between two given facts, the similarity between a certain fact and two distinct facts can be evaluated in terms of the subsumption relation between R-similar sets reasonably .

For example, let R_1 be a set of rules shown as follows.

$$\begin{aligned} R_1 = \{&\texttt{parent}(x,y) \leftarrow \texttt{father}(x,y),\\ &\texttt{parent}(x,y) \leftarrow \texttt{mother}(x,y),\\ &\texttt{family}(x,y) \leftarrow \texttt{parent}(x,y),\\ &\texttt{family}(x,y) \leftarrow \texttt{brother}(x,y),\\ &\texttt{hates}(x,y) \leftarrow \texttt{kills}(x,y),\\ &\texttt{hates}(x,y) \leftarrow \texttt{hurts}(x,y),\\ &\texttt{hates}(x,y) \leftarrow \texttt{strikes}(x,y)\} \end{aligned}$$

Let us consider the similarity of $\texttt{father}(x,y)$ to $\texttt{mother(Jim,Betty)}$ and $\texttt{brother(Tom,Joe)}$. For each atom, the following R-deducible sets are derived as

$$\begin{aligned} &\Phi(R_1,\texttt{father}(x,y)) = \{\texttt{father}(x,y),\texttt{parent}(x,y),\texttt{family}(x,y)\}\\ &\Phi(R_1,\texttt{mother(Jim,Betty)})\\ &\qquad = \{\texttt{mother(Jim,Betty)},\texttt{parent(Jim,Betty)},\\ &\qquad\quad \texttt{family(Jim,Betty)}\}\\ &\Phi(R_1,\texttt{brother(Tom,Joe)})\\ &\qquad = \{\texttt{brother(Tom,Joe)},\texttt{family(Tom,Joe)}\}. \end{aligned}$$

R-similar sets of $\texttt{father}(x,y)$ for $\texttt{mother(Jim,Betty)}$ and $\texttt{brother(Tom,Joe)}$ are as follows.

$$\begin{aligned} &\Psi(R_1,\texttt{father}(x,y),\texttt{mother(Jim,Betty)})\\ &\qquad = \{\texttt{parent}(x,y),\texttt{family}(x,y)\}\\ &\Psi(R_1,\texttt{father}(x,y),\texttt{brother(Tom,Joe)}) = \{\texttt{family}(x,y)\} \end{aligned}$$

Accordingly $\texttt{father}(x,y)$ is more similar to $\texttt{mother(Jim,Betty)}$ than $\texttt{brother(Tom,Joe)}$ with respect to R_1. This result matches our intuition very well.

2.2 Similarity between two finite sets of atoms

The similarity between two finite sets of atoms is determined by the similarity between elements of each set. In this case, we also have to consider the matching between atoms in each set. We begin with the definition of correspondence between two sets of atoms.

Definition 4 (Correspondence) *Let A and B be finite sets of atoms.* **Correspondence φ of A to B** *is defined as follows,*

1. *φ is a relation on A and B.*
2. *There is a substitution θ and for all $(\alpha,\beta) \in \varphi\theta$,*

$$\begin{aligned} &arity(\alpha) = arity(\beta),\\ &arg(\alpha,n) = arg(\beta,n) \quad (n = 1,2,\ldots), \end{aligned}$$

where $arity(\alpha)$ indicates the number of arguments of α, and $arg(\alpha,n)$ indicates the value of n-th argument of α.

3. *For all $\alpha \in A$, there is an atom β such that $(\alpha,\beta) \in \varphi$. And for all $\beta \in B$, there is an atom α such that $(\alpha,\beta) \in \varphi$.*

For example, let A_1 and B_1 be sets of atoms shown as follows.

$$\begin{aligned} A_1 &= \{\texttt{father}(x,y),\texttt{kills}(y,z)\}\\ B_1 &= \{\texttt{mother(Jim,Betty)},\texttt{hurts(Betty,Jim)}\} \end{aligned}$$

In this case, two correspondences φ_1, φ_2 of A_1 to B_1 are obtained.

$$\begin{aligned} \varphi_1 &= \{(\texttt{father}(x,y),\texttt{mother(Jim,Betty)}),\\ &\qquad (\texttt{kills}(y,z),\texttt{hurts(Betty,Jim)})\}\\ \varphi_2 &= \{(\texttt{father}(x,y),\texttt{hurts(Betty,Jim)}),\\ &\qquad (\texttt{kills}(y,z),\texttt{mother(Jim,Betty)})\} \end{aligned}$$

Definition 5 (Precedence of correspondence)
Let A and B be sets of atoms, φ_1 and φ_2 be two distinct correspondences of A to B. Then

- *For all α in A, α is similar to β_1 such that $(\alpha, \beta_1) \in \varphi_1$ than β_2 such that $(\alpha, \beta_2) \in \varphi_2$, or the similarity between α and β_1 is equal to the similarity between α and β_2 with respect to R, and*

- *There exists α in A, which is similar to β_1 such that $(\alpha, \beta_1) \in \varphi_1$ than β_2 such that $(\alpha, \beta_2) \in \varphi_2$, with respect to R,*

if and only if we say that **correspondence φ_1 precedes φ_2 with respect to R.** *For a correspondence φ of A to B, if there is no correspondence that precedes φ, we call φ a* **maximally preceding correspondence of A to B with respect to R.**

Maximally preceding correspondence represents the matching between the most similar atoms in two sets of atoms with binding variables consistently.

In the above example, φ_1 precedes another correspondence, namely, φ_2, with respect to R_1, because `father`(x, y) is more similar to `mother(Jim,Betty)` than `hurts(Betty,Jim)` and likewise `kills`(y, z) is more similar to `hurts(Betty,Jim)` than `mother(Jim,Betty)`. Therefore φ_1 is a maximally preceding correspondence of A_1 to B_1 with respect to R_1.

Definition 6 (Similarity between sets of atoms) *Let A, A', B and C be sets of atoms, φ_B be a maximally preceding correspondence of A to B with respect to R and φ_C be a maximally preceding correspondence of A' to C with respect to R. Then*

- *For all α in $A \cap A'$, α is similar to β_B such that $(\alpha, \beta_B) \in \varphi_B$ than β_C such that $(\alpha, \beta_C) \in \varphi_C$, or the similarity between α and β_B is equal to the similarity between α and β_C with respect to R, and*

- *There exists α in $A \cap A'$, which is similar to β_B such that $(\alpha, \beta_B) \in \varphi_B$ than β_C such that $(\alpha, \beta_C) \in \varphi_C$, with respect to R,*

if and only if we say that the similarity between A and B **is stronger than** *the similarity between A' and C* **with respect to R**, *denoted by*

$$[A : B] \stackrel{R}{\succ} [A' : C].$$

Now, we assume C_1 is the following set of atoms.

$$C_1 = \{\texttt{brother(Tom,Joe)}, \texttt{strikes(Joe,Mark)}\}$$

A maximally preceding correspondence of A_1 to C_1 with respect to R_1 is shown as

$$\{(\texttt{father}(x, y), \texttt{brother(Tom, Joe)}), (\texttt{kills}(y, z), \texttt{strikes(Joe, Mark)})\},$$

and therefore,

$$[A_1 : B_1] \stackrel{R}{\succ} [A_1 : C_1].$$

2.3 Formulation of analogical generalization

In this section, we proceed to formulate analogical generalization. First we give a logical consideration on analogical generalization under five conditions to generate a rule, discussing these conditions briefly.

Let τ be a non-ground atom which represents a target concept, and E be an example, that is, a set of ground atoms which is relevant to the target concept. In this case a non-ground atom is an atom containing variables and a ground atom is an atom containing no variable. We assume that E contains τ', called target instance, such that *unifiable*$(\{\tau, \tau'\})$. Let E' be a set given by removing target instance τ' from E, and E'' be a set of ground atoms deduced by $R \cup E$. Analogical generalization is formulated as follows.

Definition 7 (Analogical generalization) *Given R, E, τ, and if*

$$R \cup E' \not\vdash \tau', \tag{1}$$

then generating a rule r such that

$$R \cup E' \cup \{r\} \vdash \tau', \tag{2}$$

$$R \cup E' \cup \{r\} \textit{ is consistent, and} \tag{3}$$

$$r \textit{ satisfies the following five conditions,} \tag{4}$$

is called **analogical generalization.**

- *Selection condition*
  There is a substitution θ such that
  $$\Pi(r)\theta \subseteq E'',$$
  $$cons(r)\theta = \tau',$$
  where $\Pi(r)$ denotes a set of all atoms that constitute r.

- *Similarity condition*
  There is a rule $r'(\in R)$, provided that
  1. *There is a correspondence of $\Pi(r')$ to $\Pi(r)\theta$, which contains $(cons(r'), \tau')$ [1].*
  2. *For an arbitrary set of atoms $A(\subseteq E'')$, the following relation does not hold.*
  $$[\Pi(r') : A] \stackrel{R}{\succ} [\Pi(r') : \Pi(r)\theta].$$
  3. *For an arbitrary rule $r''(\in R)$ and an arbitrary set of atoms $A(\subseteq E'')$, the following relation does not hold.*
  $$[A : \Pi(r'')] \stackrel{R}{\succ} [\Pi(r)\theta : \Pi(r')].$$

- *Significance condition*
  For a rule r' which satisfies similarity condition [2], letting φ be a correspondence of $\Pi(r')$ to $\Pi(r)\theta$,
  $$\bigcup_{(\alpha,\beta)\in\varphi} \Psi(R, \alpha, \beta) \neq \emptyset.$$

[1] θ indicates the same substitution in selection condition.
[2] We call r' a base rule.

- *Generality condition*
  For a base rule r', letting φ be a correspondence of $\Pi(r')$ to $\Pi(r)$,

$$^{\forall}(\alpha,\beta) \in \varphi,\ \arg(\alpha, n) = \arg(\beta, n)\ (n = 1, 2, \ldots).$$

- *Applicability condition*
  For a base rule r', let φ_1 be a correspondence of $\Pi(r')$ to $\Pi(r)\theta$. Let φ_2 be a correspondence of $\Pi(r')$ to $A(\subseteq E'')$ which contains r', provided that φ_2 contains $(cons(r'), r')$. For all $\alpha \in \Pi(r')$, if $R \cup \{\alpha\} \not\vdash \beta_2$ or $\{\alpha\} \vdash \beta_2$ such that $(\alpha, \beta_2) \in \varphi_2$, $R \cup \{\beta_1\} \not\vdash \beta_2$ or $\beta_1 = \beta_2$ such that $(\alpha, \beta_1) \in \varphi_1$ has to holds.

Since there are, in general, many rules satisfying the equation (2) and (3), we have introduced the five conditions as constraints for the rule r.

Selection condition means that the rule r is generated making use of predicates which are used for representing given examples and existing rules.

Similarity condition is a condition for the purpose of generating a rule which is similar to an existing rule. A base rule, which is the most similar rule to a given example in existing rules, is selected appropriately due to this condition. Moreover, it guarantees that, with respect to the similarity, relevant atoms are extracted from the example for the selected base rule. That is, this condition is regarded as a bias depending on the domain specific knowledge.

Similarity condition is a condition for checking the validity of a base rule based on a relative comparison of the similarities between a base rule and an example, while significance condition investigates absolutely the relevance between a base rule and an example by means of R-similar set. Rules not satisfying significance condition should be regarded as absurd rules.

Generality condition removes constants which occur in an example from the generated rule. It aims at the versatility of the generated rule.

If an atom α forms a rule r and $R \cup \{\alpha\}$ is able to deduce another atom α', a rule formed by an atom α' instead of α also satisfies the equation (2) and (3). In this case, the latter rule is more applicable than the former. Applicability condition guarantees the most applicable rule can be adopted.

3 ANGEL

3.1 Procedure

This section presents ANGEL in detail. If the set of existing rules R, an example E and target concept τ are given, ANGEL generate a new rule by means of analogical generalization. We show the overview of ANGEL in Figure 1.

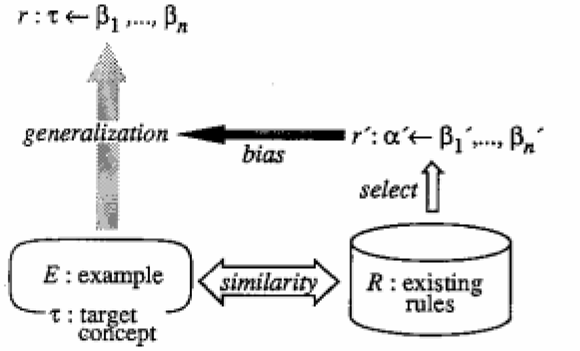

Figure 1: Overview of ANGEL

If R consists of recursive rules, R-deducible set will be infinite. Then, we assume R has no recursive rule for computing the similarity between atoms practically.

The procedure of ANGEL consists of five steps: (1) extending an example, (2) extracting atoms from the example and selecting a base rule out of the set of existing rules, (3) generalizing the extracted atoms, (4) replacing predicates, and (5) generating a rule. We show briefly each step as below.

STEP1 Extending an example

Generate a set of ground atoms which are deduced by $R \cup E$ and denote it by $\tilde{E}$. If an atom $\alpha(\in \tilde{E})$ can be deduced by $R \cup \{\alpha'\}$ $(\alpha' \neq \alpha, \alpha' \in \tilde{E})$, remove the atom α from $\tilde{E}$.

STEP2 Extracting atoms and selecting a base rule

For each rule $r' \in R$, make correspondences of $\Pi(r')$ to A which is an arbitrary subset of $\tilde{E}$. At this time, $cons(r')$ will certainly correspond to the target instance. If a set $A'(\neq A)$ such that,

$$[\Pi(r') : A'] \overset{R}{\succ} [\Pi(r') : A],$$
$$A' \subseteq \tilde{E}$$

does not exist, regard the correspondence of $\Pi(r')$ to A as a candidate of useful correspondence; otherwise abandon the set A. Note that once abandoned sets for a certain rule are never adopted for other rules.

For all candidates of useful correspondences, evaluate the similarities between subsets of an example and rules. And if a correspondence of A' to $\Pi(r'')$ such that,

$$[A' : \Pi(r'')] \overset{R}{\succ} [A : \Pi(r')],$$
$$A' \subseteq \tilde{E},$$
$$r'' \in R$$

does not exist, adopt the correspondence of A to $\Pi(r')$ as a useful correspondence.

STEP3 Generalizing atoms

Generalization is performed by turning constants to variables. As a result of STEP2, there is at least one useful correspondence φ of $\Pi(r')$, in which r' is selected out of R, to A, which is a subset of $\tilde{E}$. Now, turn constants in atoms in the set A to variables which occur at the same position of $\Pi(r')$ according to the correspondence φ.

STEP4 Replacing predicates

For each pair of atom (α, β) in φ which is a useful correspondence of $\Pi(r')$ to A, if $\Phi(R, \beta)$ contains an atom which consists of the same predicates as α, replace the predicate of β with the predicate of α. Otherwise, let S be a set of atoms in $\Phi(R, \beta)$ provided that none of whose predicates occurs in $\Phi(R, \alpha)$. Replace the predicate of β with the predicate of $\gamma (\in S)$ such that

$$^{\forall}\gamma' \in S, \quad \Phi(R, \gamma') \supseteq \Phi(R, \gamma).$$

STEP5 Generating a rule

Finally, generate a new rule r in which $cons(r)$ consists of the atom which is generalization of the target instance and $prem(r)$ consists of the atoms which are generalizations of the atoms in the set A except the target instance.

3.2 Examples and discussions

In this section, we present the two examples of learning by ANGEL. And we clarify the effectiveness of ANGEL by considering the experimental results.

First, we show a simple example in order to follow the behavior of ANGEL. A set R_2 which consists of seven existing rules defines relations of family. E_1 is an example for the target concept "$\mathtt{grandmother}(s,t)$".

$$\begin{array}{lll} R_2 = \{ & \mathtt{grandfather}(x,z) \leftarrow \mathtt{parent}(x,y), \mathtt{father}(y,z), & \cdots (r1) \\ & \mathtt{uncle}(x,z) \leftarrow \mathtt{parent}(x,y), \mathtt{brother}(y,z), & \cdots (r2) \\ & \mathtt{cousin}(x,y) & \\ & \quad \leftarrow \mathtt{parent}(x,v), \mathtt{parent}(y,w), \mathtt{brother}(v,w), & \cdots (r3) \\ & \mathtt{parent}(x,y) \leftarrow \mathtt{mother}(x,y), & \cdots (r4) \\ & \mathtt{parent}(x,y) \leftarrow \mathtt{father}(x,y), & \cdots (r5) \\ & \mathtt{family}(x,y) \leftarrow \mathtt{parent}(x,y), & \cdots (r6) \\ & \mathtt{family}(x,y) \leftarrow \mathtt{brother}(x,y) \} & \cdots (r7) \end{array}$$

$$\begin{array}{ll} E_1 = \{ & \mathtt{grandmother(Peter,Mary)}, \\ & \mathtt{mother(Paul,Mary)}, \\ & \mathtt{father(Peter,Paul)}, \\ & \mathtt{mother(Peter,Lucy)}, \\ & \mathtt{likes(Paul,Mary)}, \\ & \mathtt{engineer(Peter)}, \\ & \mathtt{student(Paul)} \} \end{array}$$

If E_1 is given, ANGEL starts to extend the example. In this case, since no atom has been deduced, the extension of E_1 is E_1 itself.

In STEP2, candidates of useful subsets of E_1 are found for the rule $r1$ as follows.

$$\begin{array}{ll} \{\mathtt{grandmother(Peter,Mary)}, & \\ \ \mathtt{father(Peter,Paul)}, & \\ \ \mathtt{mother(Paul,Mary)}\} & \cdots (s1) \\ \{\mathtt{grandmother(Peter,Mary)}, & \\ \ \mathtt{father(Peter,Paul)}, & \\ \ \mathtt{likes(Paul,Mary)}\} & \cdots (s2) \end{array}$$

In these sets, since the relation

$$[\Pi(r1) : s1] \overset{R_2}{\succ} [\Pi(r1) : s2]$$

holds, the set $s2$ is abandoned. As a result, only $s1$ are adopted as the useful set of atoms. Likewise, $s1$ is adopted for the rule $r2$. And no set of atoms is adopted for other rules $r3 \sim r7$.

Next, the similarity between $\Pi(r1)$ and $\Pi(r2)$ is evaluated. As a result, the rule $r1$ is adopted as a useful rule, because the relation

$$[s1 : \Pi(r1)] \overset{R_2}{\succ} [\ s1 : \Pi(r2)]$$

holds.

In STEP3, the generalization will be accomplished. Now, there have been the following correspondences of $\Pi(r1)$ to $s1$.

$$\begin{array}{l} \{(\mathtt{grandfather}(x,z), \mathtt{grandmother(Peter,Mary)}), \\ \ (\mathtt{parent}(x,y), \mathtt{father(Peter,Paul)}), \\ \ (\mathtt{father}(y,z), \mathtt{mother(Paul,Mary)})\} \end{array}$$

Therefore, the set of generalized atoms are obtained as follows.

$$\{\mathtt{grandmother}(x,z), \mathtt{father}(x,y), \mathtt{mother}(y,z)\} \quad \cdots (s1')$$

Next, in STEP4, predicates in $s1'$ are replaced with more applicable one. In this case, predicate `father` in $s1'$ is replaced with predicate `parent`, because predicate `parent` occurs in $\Phi(R_2, \mathtt{father}(x,y))$. While predicate `mother` in $s1'$ is not replaced, because predicate `father` never occurs in $\Phi(R_2, \mathtt{mother}(y,z))$ and atom $\mathtt{mother}(y,z)$ is the only one atom in $\Phi(R_2, \mathtt{mother}(y,z))$ except atoms in $\Phi(R_2, \mathtt{father}(y,z))$. As a result of the replacement of predicates, a set of atoms are modified as

$$\{\mathtt{grandmother}(x,z), \mathtt{parent}(x,y), \mathtt{mother}(y,z)\}. \quad \cdots (s1'')$$

In STEP5, finally, according to the above set $s1''$, the following new rule is generated and added to R_2.

$$\mathtt{grandmother}(x,z) \leftarrow \mathtt{parent}(x,y), \mathtt{mother}(y,z) \quad \cdots (r8)$$

The rule $r8$ satisfies the requirement for analogical generalization given at Definition 7, and it is just appropriate rule about the target concept. In this case, good learning has been performed, because the rule which is closely similar to the rule for target concept is in the existing knowledge base.

In rule based systems, generally, the lack of rules causes either interruptions or mistakes on inference. ANGEL is useful for such a situation, because it is possible to continue inference by generating new rules from given examples.

Next we show an example of acquiring rules for the system for parsing simple English sentences. The target system is capable of parsing English sentences by means of syntactic rules shown as Figure2. In this system a sentence is treated as a list. For example the sentence "*The sun rises in the east*" is represented as the list,

```
[the, sun, rises, in, the, east]
```

And

```
noun_phrase([the,sun,rises,in,the,east],
            [rises,in,the,east])
```

indicates that [the, sun] is noun phrase. The system examines whether or not a given sentence is grammatically valid by a backward chaining inference by means of the syntax rules.

$\mathtt{sentence}(s,e) \leftarrow \mathtt{noun\_phrase}(s,v_1), \mathtt{verb\_phrase}(v_1,e).$
$\mathtt{sentence}(s,e) \leftarrow \mathtt{noun\_phrase}(s,v_1), \mathtt{verb\_phrase}(v_1,v_2),$
$\quad\quad \mathtt{prepositional\_phrase}(v_2,e).$
$\mathtt{sentence}(s,e) \leftarrow \mathtt{present\_progressive}(s,e).$
$\mathtt{sentence}(s,e) \leftarrow \mathtt{present\_passive\_voice}(s,e).$
$\mathtt{sentence}(s,e) \leftarrow \mathtt{present\_perfect}(s,e).$
$\mathtt{noun\_phrase}(s,e) \leftarrow \mathtt{determiner}(s,v_1), \mathtt{noun}(v_1,e).$
$\mathtt{noun\_phrase}(s,e) \leftarrow \mathtt{noun}(s,e).$
$\mathtt{prepositional\_phrase}(s,e) \leftarrow \mathtt{preposition}(s,v_1),$
$\quad\quad \mathtt{noun\_phrase}(v_1,e).$
$\mathtt{verb\_phrase}(s,e) \leftarrow \mathtt{verb}(s,e).$
$\mathtt{verb\_phrase}(s,e) \leftarrow \mathtt{verb}(s,v_1), \mathtt{noun\_phrase}(v_1,e).$
$\mathtt{present\_progressive}(s,e) \leftarrow \mathtt{noun\_phrase}(s,v_1),$
$\quad\quad \mathtt{present\_BE}(v_1,v_2), \mathtt{present\_participle}(v_2,e)$
$\mathtt{present\_progressive}(s,e) \leftarrow \mathtt{noun\_phrase}(s,v_1),$
$\quad\quad \mathtt{present\_BE}(v_1,v_2), \mathtt{present\_participle}(v_2,v_3),$
$\quad\quad \mathtt{noun\_phrase}(v_3,e)$
$\mathtt{verb}(s,e) \leftarrow \mathtt{BE}(s,e).$
$\mathtt{verb}(s,e) \leftarrow \mathtt{main\_verb}(s,e).$
$\mathtt{verb}(s,e) \leftarrow \mathtt{present\_verb}(s,e).$
$\mathtt{verb}(s,e) \leftarrow \mathtt{past\_verb}(s,e).$
$\mathtt{BE}(s,e) \leftarrow \mathtt{present\_BE}(s,e).$
$\mathtt{BE}(s,e) \leftarrow \mathtt{past\_BE}(s,e).$
$\mathtt{main\_verb}(s,e) \leftarrow \mathtt{present\_main\_verb}(s,e).$
$\mathtt{main\_verb}(s,e) \leftarrow \mathtt{past\_main\_verb}(s,e).$
$\mathtt{present\_verb}(s,e) \leftarrow \mathtt{present\_BE}(s,e).$
$\mathtt{past\_verb}(s,e) \leftarrow \mathtt{past\_BE}(s,e).$
$\mathtt{present\_verb}(s,e) \leftarrow \mathtt{present\_main\_verb}(s,e).$
$\mathtt{past\_verb}(s,e) \leftarrow \mathtt{past\_main\_verb}(s,e).$
$\mathtt{auxiliary\_verb}(s,e) \leftarrow \mathtt{present\_auxiliary\_verb}(s,e).$
$\mathtt{auxiliary\_verb}(s,e) \leftarrow \mathtt{past\_auxiliary\_verb}(s,e).$
$\mathtt{participle}(s,e) \leftarrow \mathtt{present\_participle}(s,e).$
$\mathtt{participle}(s,e) \leftarrow \mathtt{past\_participle}(s,e).$
$\mathtt{determiner}(s,e) \leftarrow \mathtt{THE}(s,e).$
$\mathtt{noun}(s,e) \leftarrow \mathtt{SUN}(s,e).$
$\mathtt{noun}(s,e) \leftarrow \mathtt{EAST}(s,e).$
$\mathtt{noun}(s,e) \leftarrow \mathtt{DOOR}(s,e).$
$\mathtt{noun}(s,e) \leftarrow \mathtt{HER}(s,e).$
$\mathtt{noun}(s,e) \leftarrow \mathtt{HE}(s,e).$
$\mathtt{noun}(s,e) \leftarrow \mathtt{I}(s,e).$
$\mathtt{noun}(s,e) \leftarrow \mathtt{HOMEWORK}(s,e).$
$\mathtt{present\_main\_verb}(s,e) \leftarrow \mathtt{HAVE}(s,e).$
$\mathtt{present\_main\_verb}(s,e) \leftarrow \mathtt{RISES}(s,e).$
$\mathtt{present\_auxiliary\_verb}(s,e) \leftarrow \mathtt{HAVE}(s,e).$
$\mathtt{present\_BE}(s,e) \leftarrow \mathtt{IS}(s,e).$
$\mathtt{past\_participle}(s,e) \leftarrow \mathtt{CLOSED}(s,e).$
$\mathtt{past\_participle}(s,e) \leftarrow \mathtt{RESPECTED}(s,e).$
$\mathtt{past\_participle}(s,e) \leftarrow \mathtt{FINISHED}(s,e).$
$\mathtt{preposition}(s,e) \leftarrow \mathtt{IN}(s,e).$
$\mathtt{preposition}(s,e) \leftarrow \mathtt{BY}(s,e).$

Figure 2: A part of rules in existing knowledge base

As Figure2 indicates, initially, the rule to define syntax about the present passive voice is insufficient. Then we have tried to generate a lacking rule by ANGEL.

For the target concept "$\mathtt{present\_passive\_voice}(s,e)$", we have given the following example E_2 to ANGEL.

```
E2 = { present_passive_voice([the,door,is,closed],[]),
       THE([the,door,is,closed],[door,is,closed]),
       DOOR([door,is,closed],[is,closed]),
       IS([is,closed],[closed]),
       CLOSED([closed],[])}
```

Firstly, the given example E_2 has been extended to the following set $\bar{E}_2$.

```
Ē2 = { present_passive_voice([the,door,is,closed],[]),
       THE([the,door,is,closed],[door,is,closed]),
       DOOR([door,is,closed],[is,closed]),
       IS([is,closed],[closed]),
       CLOSED([closed],[]),
       noun_phrase([the,door,is,closed], [is,closed]),
       sentence([the,door,is,closed],[closed])}
```

Then, the useful correspondence has been found as follows by using a rule for "present_progressive" as a base rule.

```
{(present_progressive(s,e),
  present_passive_voice([the,door,is,closed],[])),
 (noun_phrase(s,v1),
  noun_phrase([the,door,is,closed],[is,closed])),
 (present_BE(v1,v2), IS([is,closed],[closed])),
 (present_participle(v2,e),CLOSED([closed],[]))}
```

As a result, we have confirmed that ANGEL generates the following one rule successfully.

$$\mathtt{present\_passive\_voice}(s,e) \leftarrow \mathtt{noun\_phrase}(s,v_1), \mathtt{present\_BE}(v_1,v_2), \mathtt{past\_participle}(v_2,e) \quad \cdots (r9)$$

The generated new rule $r9$ is added to the knowledge base.

Again we have given an example sentence "A mouse is caught by a cat." for the same target concept.

In this case, two distinct rules $r10$ and $r11$ are generated by using the identical base rule in the existing knowledge base.

$$\mathtt{present\_passive\_voice}(s,e) \leftarrow \mathtt{noun\_phrase}(s,v_1), \mathtt{present\_BE}(v_1,v_2), \mathtt{past\_participle}(v_2,v_3), \mathtt{prepositional\_phrase}(v_3,e) \quad \cdots (r10)$$

$$\mathtt{present\_passive\_voice}(s,e) \leftarrow \mathtt{sentence}(s,v_1), \mathtt{participle}(v_1,v_2), \mathtt{preposition}(v_2,v_3), \mathtt{noun\_phrase}(v_3,e) \quad \cdots (r11)$$

Like the above, ANGEL sometimes generates several rules for one example. It is now important to examine whether each of the generated rules is appropriate. For instance, The rule $r10$ is a suitable rule, whereas the rule $r11$ is obviously strange. The reason for this is none of the rules in the existing knowledge base are really similar to the given example. Since atom $\mathtt{noun\_phrase}(v_3,e)$ in selected base rule

$$\mathtt{present\_progressive}(s,e) \leftarrow \mathtt{noun\_phrase}(s,v_1), \mathtt{present\_BE}(v_1,v_2), \mathtt{present\_participle}(v_2,v_3), \mathtt{noun\_phrase}(v_3,e)$$

corresponds to atom prepositional_phrase(v_3, e) in the rule $r10$ and atom noun_phrase(v_3, e) in the rule $r11$ (namely, the given example is regarded as the sentence consisting of some phrases and noun_phrase), the similarity between the base rule and the rule $r11$ are stronger than the one between the base rule and the rule $r10$ in respect of these atoms.

Next, we have supplied a sentence "He was killed by them. " to attempt to generate a rule for another target concept past_passive_voice(s, e). ANGEL could generate a new rule $r12$ by employing a rule $r10$ generated just now.

$$\begin{aligned}\texttt{past\_passive\_voice}(s,e) \leftarrow\ & \texttt{noun\_phrase}(s,v_1),\\ & \texttt{past\_BE}(v_1,v_2),\\ & \texttt{past\_participle}(v_2,v_3),\\ & \texttt{prepositional\_phrase}(v_3,e)\end{aligned} \qquad \cdots(r12)$$

In this case, since an appropriate base rule, which does not exist initially, has occurred in knowledge base, a good rule is generated accurately by selecting it. ANGEL is capable of growing knowledge base gradually by employing rules generated by ANGEL itself as base rules.

Let us discuss the computational complexity of ANGEL. In order to evaluate the similarity between atoms, ANGEL has to compute deductive closures of each of the atoms. And the similarities between atoms in arbitrary correspondences have been estimated to find the most suitable pair of the atoms in the given example and the base rule. Therefore, procedure of ANGEL may be expensive as a whole, although hypothesis space to be considered is small. In fact, as a result of implementing ANGEL on Sun SPARC Station2 with SICStus Prolog, it took a few minutes to generate a English syntax rule.

The approach evaluating similarities between atoms based on their deductive closures is theoretically interesting, but it may not be practical. For the purpose of practical learning, some restrictions on either forms of the background knowledge or the hypothesis language are required like Muggleton's GOLEM[Muggleton 1990]. We think we will have to improve the practicability of ANGEL in the near future.

4 Related works

In this section, we characterize ANGEL from a viewpoint of general machine learning framework.

ANGEL belongs to the category of learning from examples, in the sense that it generates new rules by generalizing given examples. In inductive learning methods, generally, pre-defined generalization rules are used for generalizing examples. ANGEL also uses three kinds of generalization rules corresponding to dropping condition rule, turning constants to variables rule and constructive generalization rule based on logical implications [Michalski 1983], all of them are considered as the primary generalization rules in learning from examples. However, ANGEL differs from the ordinary inductive learning methods in using the existing rules as the bias. That is, ordinary inductive learning uses no existing rules, even if so, it uses them for the constructive induction. On the other hand, ANGEL employs the similarity between the existing rules and the given example in order to drop conditions, so it can reduce the hypothesis space extremely.

ANGEL is related to inductive logic programming (ILP), because it generates rules represented as Horn clauses by induction. ILP is also capable of learning new rules with reference to existing rules. Both Muggleton and Buntine's CIGOL[Muggleton and Buntine 1988] and Wirth's LFP2[Wirth 1989], which are typical examples of ILP system, use operators based on inverting resolution to augment incomplete clausal theories. The difference between these systems and ANGEL is the way of employing existing background knowledge. That is, in both of their systems, background knowledge is not employed as biases at all. In fact, rules can be acquired under no background knowledge. Therefore the interaction between user and system is inevitable in their systems to derive reasonable rules. Whereas, ANGEL employs background knowledge as a bias. A given example is generalized through mapping a structure of a rule in existing knowledge base. It provides a strong restriction for induction and serves to generate a few useful new rules.

ANGEL evaluates a similarity between existing rule and a given example to learn a new rule. Therefore it can also be regarded as a kind of method for learning by analogy. Davies and Russell [1987] have defined, in their paper, reasoning by analogy as the process of inferring that a property Q holds of a particular situation T (called the target) from the fact that T shares a property P with another situation S (called the source) that has property Q. In analogy, it is very important to match between the target and the source. Similarly, in ANGEL, the matching between existing rules and a given example, which is called correspondence in this paper, must be found successfully. Now we compare ANGEL with several methods with respect to the way of matching.

Haraguchi and Arikawa [1986] have formalized the reasoning by analogy on a deduction system. In their method, the domain for reasoning is represented by a set of definite clauses, and the similarity between objects is defined as the identity of predicates. Therefore the matching is performed by pairing the atoms which are described with the same predicate. On the other hand, ANGEL finds a correspondence between atoms based on their similarities, that is, it will not require identity of predicates. And it enables ANGEL to generate completely novel rules.

Recently, Arima [1991] has analyzed analogy from the point of logical relevance. His formulation is based on the idea as follows.

1. The property to be projected from the source to the target must be justified.
2. The similarities, which means the properties shared by both the source and the target, should be formed by the minimum justifications.

Unlike ANGEL, the shared properties must be represented by the same predicates both with the source and with the target.

Gentner [1983] has also developed a method, called Structure Mapping, for the matching between the target and the source. In her method, first an atom is matched with another atom, when both of them are described with the same predicates, and next, the object in each atom is matched. And the process of the matching is repeated based on newly matched objects. ANGEL is similar to Structure Mapping, because the matching between atoms is achieved based on the matched objects. However, there are the following two differences between them.

1. Although Structure Mapping requires the identity to several kinds of predicates (e.g. *greater*, *cause*, etc.) in order to match between atoms, ANGEL will not require the identity of predicates at all.

2. In Structure Mapping, the similarity between descriptions is defined by the identification of predicates and the number of matched descriptions. On the other hand, in ANGEL, it is defined as the subsumption between deductive closures of atoms based on the logical consideration.

ANGEL is also related to both the explanation-based learning (EBL)[Mitchell *et al.* 1986] and Russell's single-instance generalization (SIG)[Russell 1987], because all of them are capable of learning from one example and background knowledge. However, EBL has to need completeness for background knowledge, so rules produced by EBL are limited to ones which are deducible from background knowledge. In this sense, EBL cannot generate really new rules. SIG requires weak background knowledge, called determinations, in stead of complete one. That is, it can learn rules under comparatively insufficient background knowledge in contrast to EBL. Properly new rules cannot, however, be generated, because it does not deal with non-deductive reasoning.

5 Conclusion

This paper has described an approach to learning from an example by analogical generalization.

The notable features of ANGEL are shown as follows.

1. ANGEL is able to generate a new rule from a given single example by analogical generalization.

2. A similarity between an existing rule and an example can be evaluated a similarity between atoms forming each of them.

3. A similarity between atoms is defined based on the subsumption relation between deductive closures of atoms, and it enables to compute similarities formally.

Through the experiment for the domain of parsing English sentences, we have confirmed that ANGEL is useful for acquiring knowledge on knowledge based systems.

In this paper, from the inductive learning point of view, we have highlighted the method to generate a new rule from a given example. The definition of similarity introduced here is not specific for inductive learning. We plan to apply this idea to other various reasoning paradigms (e.g. ordinary analogical reasoning, deductive reasoning and so on) to improve performance and applicability of them.

This work was supported partly by the Grant-in-Aid for scientific research from the Ministry of Education.

References

[Mitchell 1977] T. M. Mitchell: "Version spaces: a candidate elimination approach to rule learning", Proc. of 5th IJCAI, pp.305–310 (1977).

[Winston 1980] P. H. Winston: "Learning and reasoning by analogy", Comm. of ACM, Vol.23, No.12, pp.689–703 (1980).

[Dietterich and Michalski 1983] T. G. Dietterich and R. S. Michalski: "A comparative review of selected methods for learning from examples", Machine learning, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), Morgan Kaufmann, pp.41–81 (1983).

[Michalski 1983] R. S. Michalski: "A theory and methodology of inductive learning", Machine learning, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), Morgan Kaufmann, pp.83–134 (1983).

[Gentner 1983] D. Gentner: "Structure-mapping: A theoretical framework for analogy", Cognitive science, Vol. 7, pp.155–170 (1983).

[Haraguchi and Arikawa 1986] M. Haraguchi and S. Arikawa: "A formulation of analogical reasoning and its realization", Journal of JSAI, Vol. 1, No. 1, pp.132–139 (1986) (in Japanese).

[Mitchell *et al.* 1986] T. M. Mitchell, R. M. Keller and S. T. Kedar-Cabelli: "Explanation-based generalization: A unifying view", Machine Learning, Vol.1 No.1, pp.47–80 (1986).

[Davies and Russell 1987] T. R. Davies and S. J. Russell: "A logical approach to reasoning by analogy", Proc. 10th IJCAI, pp.264–270 (1987).

[Russell 1987] S. J. Russell: "Analogy and single-instance generalization", Proc. 4th Int'l Machine Learning Workshop, pp.390–397 (1987).

[Muggleton and Buntine 1988] S. Muggleton and W. Buntine: "Machine invention of first-order predicates by inverting resolution", Proc. of 4th Int'l Machine Learning Workshop, pp.390–397 (1988).

[Wirth 1989] R. Wirth: "Completing logic programs by inverse resolution", Proc. of 4th European Working Session on Learning, pp.239–250 (1989).

[Muggleton 1990] S. Muggleton and C. Feng: "Efficient induction of logic programs", Proc. of 1st Int'l Workshop on Algorithmic Learning Theory, pp.368–381 (1990).

[Arima 1991] J. Arima: "A logical analysis of relevancy in analogy ", Proc. of 5th Annual Conf. of JSAI, pp.235–238 (1991) (in Japanese).

[Mori *et al.* 1991] T. Mori, T. Ohkawa, N. Babaguchi and Y. Tezuka: "Learning based on similarity between rules and examples", IPS Japan research report on artificial intelligence, 74-6, pp.49–57 (1991) (in Japanese).

[Ohkawa *et al.* 1991] T. Ohkawa, T. Mori, N. Babaguchi and Y. Tezuka: "Class directed generalization", Proc. of 3rd SCAI, pp.266–276 (1991).