Automatic Construction of Rule-based Trees for Conceptual Retrieval

Minkoo Kim†, Fenghua Lu, and Vijay V. Raghavan
The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504, USA
(mkim, raghavan)@cacs.louisiana.edu

Abstract

Many intelligent retrieval approaches have been studied to bridge the terminological gap existing between the way in which users specify their information needs and the way in which queries are expressed. One of the approaches, called Rule Based Information Retrieval by Computer (RUBRIC), uses production rules to capture user query concepts (or topics). A set of related production rules is represented as an AND/OR tree, called a rule-based tree. One of the main problems in this approach is how to construct such rules that can capture user query concepts. This paper provides a logical framework semantically essential to defining the rules for the user query concepts, and proposes a way to automatically construct rule-based trees from typical thesauri. Experiments performed on small collections with a domain-specific thesaurus show that the automatically constructed rules are more effective than hand-made rules in terms of precision.

1. Introduction

Many intelligent retrieval systems have been studied [2, 3, 8, 9, 12, 13]. However, only a few systems have been used in the real domains. There are several reasons why these systems are not really being used in the world. One of the main reasons is that some intelligent retrieval systems ignore much of the conceptual information that human beings have acquired by experience, while the other systems pay much attention to representing conceptual information even though the acquisition of that knowledge for use by the systems is equally challenging. One of the pioneering studies to integrate information retrieval and artificial intelligence techniques is RUBRIC [12] that uses production rules to define a hierarchy of retrieval subtopics. RUBRIC allows the definition of detailed queries starting at a conceptual level.

However, RUBRIC still has some drawbacks when it comes to real world applications. In this paper, we point out some problems in RUBRIC and attempt to refine the ideas of RUBRIC. One major problem is the lack of semantics for the rules adopted in RUBRIC. The lack of the semantics often results in confusion and may lead to the definition of unintended rules. Another problem in RUBRIC is about knowledge acquisition. Users would prefer to retrieve documents of interest without having to define rules for their queries. This implies that a lot of predefined rules are needed. To solve these problems, we provide a logical semantics for the rules used in RUBRIC, and propose a way to automatically construct rules from typical thesauri.

2. The semantics of rules in RUBRIC

Before defining the semantics for the rules in RUBRIC, we need to establish a logical framework for retrieval models. For this purpose, we investigate and review some logical approaches [4, 15]. Based on these approaches, we assume that terms are 1-place predicates and documents are objects in the retrieval world. For example, consider formula \(\text{computer}(d)\). In the retrieval world, this formula is true if the document denoted by \(d\) is indexed by term \(\text{computer}\). In this section, we mainly focus on the meaning of symbol \(\Rightarrow\) in the form of \(\alpha \Rightarrow \beta\) for representing rules in RUBRIC.

2.1. Material implication for representing rules

We, first, consider an example of rule used in RUBRIC as follows: \(\text{baseball-championship} \Rightarrow \text{event}\). This rule

---

* This research was supported in part by the U.S. Department of Energy. Granted No. DE-FG02-97ER1220.
* On leave from the Department of Computer Engineering in Ajou University, Korea.
says that baseball championship is an event. One possible interpretation for symbol \( \Rightarrow \) is the material implication of the Boolean logic \( \supset \) having the property: \( \alpha \supset \beta \equiv \neg \alpha \lor \beta \) [11].

For theoretical completeness, we use the material implication to represent the rules, although it is not suitable to the retrieval domain. Using the material implication, we can represent it as

\[ \forall x \text{ baseball-championship}(x) \supset \text{event}(x). \]

Intuitively, this formula says that if a document is indexed by term baseball-championship, then it can be indexed by term event. For the notational convenience, we will simply represent the above formula as

baseball-championship \( \Rightarrow \) event

RUBRIC allows us to represent degrees of certainty about rules with a real number in the interval \([0, 1]\). For example, consider the following rule used in RUBRIC.

\[ \text{baseball} \& \text{championship} \Rightarrow \text{baseball-championship (0.9)} \]

In order to represent the rule using the material implication, we need to extend the classical logic with degrees of certainty. In this paper, we only give the definitions for the logical connectives NOT (\( \neg \)), AND (\( \land \)), and OR (\( \lor \)), as follows. In the below, \( t \) and \( s \) are propositions, and \( \omega, \xi, \) and \( \psi \) are degrees of certainty.

\[ \begin{align*}
- \xi(\omega) & \text{ iff } t(\psi), \text{ where } \omega = 1 - \psi \\
\xi \land s(\omega) & \text{ if } t(\xi) \text{ and } s(\psi), \text{ where } \omega = \min(\xi, \psi) \\
\xi \lor s(\omega) & \text{ if } t(\xi) \text{ or } s(\psi), \text{ where } \omega = \max(\xi, \psi)
\end{align*} \]

Now, using the material implication we can represent the above rule as follows.

\[ (\text{baseball} \land \text{championship}) \Rightarrow \text{baseball-championship (0.9)} \]

In RUBRIC, multiple rules are allowed about the same topic. For example, baseball-championship \( \Rightarrow \) event murder-event \( \Rightarrow \) event

In this case, RUBRIC uses each rule as an equally valid alternative definition. This means that there is an implicit OR. Using our notation, we can represent it as follows.

\[ (1) \text{ (baseball-championship} \lor \text{ murder-event}) \Rightarrow \text{event} \]

We can also derive formula (1) from the above two rules as follows. Since the existence of two facts in a knowledge base means the conjunction of the two facts, we can represent the two rules as in formula (2). Actually, formula (1) and formula (2) are logically equivalent.

\[ (2) \text{ (baseball-championship} \Rightarrow \text{event) } \land \text{ (murder-event} \Rightarrow \text{event)} \]

With the degrees of certainty, let us reconsider the above two rules as follows.

baseball-championship \( \Rightarrow \) event (0.9)
murder-event \( \Rightarrow \) event (0.8)

Using our notation, we can represent these rules in formula (3). By the definition of logical connectives, we can also get formula (4) from (3). The degree of certainty of this formula is 0.8 that is the minimum of 0.8 and 0.9. However, formula (4) loses the degrees of certainty for the original rules. In this sense, formula (3) is a better interpretation than formula (4).

\[ (3) \text{ (baseball-championship} \Rightarrow \text{event) (0.9) } \land \text{ (murder-event} \Rightarrow \text{event) (0.8) } \]

\[ (4) \text{ ((baseball-championship} \lor \text{ murder-event)} \Rightarrow \text{event) (0.8) } \]

In fact, the semantics of rules used in RUBRIC is not clearly mentioned in the original work [12]. Let us consider the following example that was given in [12]

(5) team \( \Rightarrow \) World-Series

One possible interpretation of this rule is formula (6). By the logical equivalence, we can also represent it as formula (7).

\[ (6) \text{ (team} \lor \text{ event) } \Rightarrow \text{World-Series} \]

\[ (7) \text{ (team} \Rightarrow \text{World-Series) } \land \text{ (event} \Rightarrow \text{World-Series) } \]

With the degrees of certainty, let us reconsider rule (5), for example as in rule (8). We note that a rule like rule (8) was actually used in some previous works [1]. One possible interpretation for it is formula (9), as mentioned before.

\[ (8) \text{ team (0.8) } \lor \text{ event (0.7) } \Rightarrow \text{World-Series} \]

\[ (9) \text{ (team} \Rightarrow \text{World-Series) (0.8) } \land \text{ (event} \Rightarrow \text{World-Series) (0.7) } \]

Someone might consider another interpretation as in formula (10) for rule (8).

\[ (10) \text{ team (0.8) } \lor \text{ event (0.7)) } \Rightarrow \text{World-Series} \]

It is theoretically permissible to consider the interpretation for formula (8). However, it is unusual to assign a specific degree of certainty to the left-hand side of a rule, instead of assigning it to the rule as a whole.

---

1. We will introduce a relevant implication later.
2. For the full version of non-classical logic, see [15].
2.2. Relevant implication for representing rules

In the previous section, we use the material implication for representing rules used in RUBRIC. However, the material implication shows some irrelevant properties. One of the properties is that false implies anything [10]. More practically, it is also difficult to get the reasonable degrees of certainty during the inference processing. For example, consider formula (11) and proposition PENTIUM with degree of certainty 0.6. By the definition of the material implication, we can convert formula (11) to formula (12). Since the degree of certainty for proposition PENTIUM is 0.6, the degree of certainty for proposition processor, denoted by cp, can be computed as in Max(1-0.6, cp) = 0.8. That is, cp = 0.8. Intuitively, the result is not plausible. It should be less than or equal to 0.6.

(11) PENTIUM ⊃ processor (0.8)
(12) ¬PENTIUM v processor (0.8)

Although the material implication is theoretically complete, we need a more practical notion of implication. We introduce a new implication, denoted by →, that can be used to represent rules. The implication → is assertion rather than logically definable. This implies that even if α ⊃ β does not hold, someone can assert α → β. The following is the syntax for defining rules.

<rule> ::= <pattern> → <concept> [degree-of-certainty]
<pattern> ::= <literals> | <pattern> & <or-operator> <pattern>
<concept> ::= <term>
<literals> ::= 0 | 1
<degree-of-certainty> ::= (real value between 0 and 1)
<and-or-operator> ::= ∧, ∨

A degree of certainty can be assigned to a rule, but one is not allowed to assign a degree of certainty to a component of a rule. If the degree of certainty is omitted, 1 is assumed.

For plausible inference processing, we provide a semantic constraint, given in Axiom 1. This axiom can be used as a generalized modus ponens inference rule.

Axiom 1. (Generalized Monus Ponens)
If t → s (ξ) and t (ψ), then s (ω), where ω = ξ * ψ

For example, consider rule (13). Suppose mammal is believed with degree of certainty 0.9 and thing-to-have-a-trunk is believed with degree of certainty 0.8. Then proposition elephant has degree of certainty 0.72 = Min(0.9, 0.8)*0.9.

(13) mammal ∧ thing-to-have-a-trunk → elephant (0.9)

We consider the problem of multiple rules, again. Suppose that we have another definition for concept elephant, as in rule (14), and that mammal, thing-to-have-a-trunk, and thing-to-have-a-long-nose are believed with degrees of certainty 0.9, 0.8, and 0.7, respectively. Then, from rule (13) we can get proposition elephant with 0.72, and from rule (14) we can derive proposition elephant with 0.56 = min(0.9, 0.7)*0.8. What is the degree of certainty for proposition elephant? Since people prefer more certain knowledge, we would better choose 0.72 as the degree. This is a kind of maximum operation.

(14) mammal ∧ thing-to-have-a-long-nose
     → elephant (0.8)

3. Defining rules from thesauri

In previous sections, we propose the semantics of rules that describe what we want to retrieve at a conceptual level. However, for developing conceptual retrieval systems in practice, it is not sufficient to provide the semantics for defining concepts. If users should define the rules whenever they want to retrieve some documents, it would be too tedious to use the retrieval systems. To address this problem, we are attempting to utilize the knowledge in thesauri.

Before extracting knowledge from thesauri, we distinguish between different types of rules that are used in defining concepts, and extract the knowledge according to the types. Suppose that a rule has a form of α → β with degree of certainty 0. Then, we can consider at least three types of rules as follows.

Type 1 (Specific Concept): If (α ⊃ β) ∧ (α c β (ω)), then α → β (ω)
Type 2 (General Concept): If (α c β) ∧ (α ⊃ β (ω)), then α → β (ω)
Type 3 (Possible Concept): If (α ⊃ β (ω)) v (α c β (ω)), then α → β (ω)

If α → β (ω) is type 1, α is more specific than concept β. For example, consider rule (15).

(15) elephant v dog v horse → mammal (0.7)

Suppose that some people believe that mammals mainly consist of elephants, dogs, and horses. Since proposition elephant v dog v horse ⊃ mammal is true, if they accept proposition elephant v dog v horse ⊃ mammal with degree of certainty 0.7 then they can make rule (15).

If α → β (ω) is type 2, α is more general than concept β. For example, consider rule (16). Since people usually believe that elephants have a trunk, it can be a reasonable conjecture that a mammal to have a trunk is an elephant.

(16) mammal ∧ thing-to-have-a-trunk → elephant (0.7).

If a rule is neither type 1 nor type 2, we might say that α is possibly related to concept β. We call this kind of
rule type 3. For example, consider rule (17). There is no
subsumption relation between concepts Scientist and
Science. But there are some relations between the two
concepts.

(17) Scientist → Science (0.4)

Now, let us look into the structure of a typical thesaurus.
A typical thesaurus usually has the following fields:
BT(Broad Term), NT(Narrow Term), RT(Related Term),
and USE. Field USE represents the terms to be used
instead of the given term with the almost same meaning.
For a given term, a thesaurus could have a list of terms for
each field. Consider a part of content in “fire ecology”
thesaurus [16] for term “Habitat types” and term “Coastal
plain.”

Descriptor: Habitat types
USE: Plant associations, Vegetation types
NT: Balds, Barrier islands, Bogs,
Bottomland hardwoods, Coastal plain,....
RT: Community ecology, Forest types
....

Descriptor: Coastal plain
BT: Habitat types
....

From NT fields in the above example, we can construct
type 1 rules as follows, although we do not yet worry
about assigning a degree of certainty to each rule.

Balds → Habitat type; Barrier islands → Habitat types;
Bottomland hardwoods → Habitat types;
Coastal plain → Habitat types;....

More generally, we can construct type 1 rules using a
disjunction of NT terms, for example Balds ∨ Barrier
islands → Habitat types. We can further extend construct
type 1 rules with a conjunction of RT terms. For example,
Balds ∨ Barrier islands ∧ Community ecology → Habitat
types.

Using a conjunction of BT terms, we can construct type
2 rules, for example Habitat types → Coastal plain, since
Habitat types ⊆ Coastal plain. Using a conjunction with
some RT terms, we can extend type 2 rules if the RT terms
subsume the given term in a conceptual sense. However,
since it is almost impossible to automatically detect
whether RT terms subsume other terms in a typical
thesaurus, we rather use syntactic rules, called AND
rules, that consist of conjunctions of BT terms and RT terms
without considering the subsumption relationship.

For type 3 rules, we adopt another category of
syntactic rules, OR rules, that consist of disjunctions of
NT terms and RT terms. Although USE terms can be used
for constructing type 3 rules, they are used implicitly,
rather than explicitly, in our retrieval system.

In summary, using the fields of a thesaurus NT, BT, and
RT, we propose to construct conceptual rules for a given
term as follows. The degrees of certainty associated with
the rules below can be experimentally decided.

Type 1 Rules: <disjunction-of-some-NT-terms>
→ <given-term> (ω₁)
<disjunction-of-some-NT-term> ∧
<conjunction-of-some-RT-terms>
→ <given-term> (ω₂)

Type 2 Rules (AND Rules):
<conjunction-of-some-BT-terms> → <given-term> (ω₃)
<conjunction-of-some-BT-term> ∧
<conjunction-of-some-RT-terms>
→ <given-term> (ω₄)

Type 3 Rules (OR Rules):
<disjunction-of-some-NT-term> ∨
<disjunction-of-some-RT-terms>
→ <given-term> (ω₅)

4. Spreading activation for choosing most
interesting terms

When the system defines concepts in a form of Type 1,
Type 2, or Type 3, it uses terms in NT field, BT field, and
RT field. Since these fields usually have a lot of terms, the
system has to select some of them in order to define the
categories that users really want. For this purpose, we are
attempting to select only the terms that user is likely to be
interested in. We assume that the terms used in concepts
previously defined for a user are interesting to the user.
We call such a set of the interesting terms the user's view.
When the system defines Type 1 and Type 3 (or Type 2)
rules in terms of NT (or BT) terms and RT terms, it will
only use the NT terms (or BT terms) and RT terms that are
related with some terms in the view. For this purpose,
we consider a function, find(t, V), such that if there exists
a path from term t to a term v in view V, it returns a
positive numerical weight, w = degree-of-closeness(t, v).
Otherwise it returns 0. We compute the degree of
closeness using the spreading activation technique in
theses [5, 6, 7, 14]. Figure 1 shows how the spreading
activation works. To do the spreading activation, we can
consider the thesaurus as a graph, where each term is a
node and each relation is a link. Given a term (node), we
want to know whether the node reaches to some node in
the view through the spreading activation.

Since the size of the set of terms to be explored in the
spreading activation increases exponentially, we will use
three kinds of constraints on spreading activation [5]. The first one is a distance constraint. It says that the activation should cease at a distance of some number of links from the starting node. The second one, called fan-out constraint, is that whenever we reach the degree of closeness below some threshold, we stop the spreading activation. The last one, called path endorsement, says that only some of the relations are allowed to be used for the spreading activation. For example, we may allow to use, NT and RT, but not BT for Type 1 and Type 3 rules.

In order to associate the constraints with the function $find(t, V)$, we add parameters $w$ and $d$ to keep track of the degree of certainty and the depth of spreading activation level, respectively. Since there are possibly multiple paths from node $t$ and a node in view $V$, we have to find the path with the highest degree of certainty. We construct function $find(t, V, w, d)$ as follows. In the function, $BT(t)$, $NT(t)$, and $RT(t)$ are a set of BT terms, a set of NT terms, and a set of RT terms for term $t$, respectively. Depending on the path endorsement, we can exclude some terms from $s$. Although this function employs a kind of depth-first search, in order to efficiently compute the function one could devise some kind of heuristic function. However, we leave this problem open.

```
find(t, V, w, d)
{
    if t is in V then return (w);
    /* Path endorsement can be applied. */
    s = NT(t) U RT(t) U BT(t);
    wmax = 0;
    for x in s {
        w = w * weight-value-of(t, x);
        if distance and fan-out constraints are satisfied
            wmax = max(wmax, find(x, v, w, d+1));
    }
    return (wmax);
}
```

5. Experimental test and evaluation

In order to test our model, we choose a fire ecology thesaurus and database [16]. We let an expert on "ecology" create rules for thirty concepts that the expert is interested in. Using these rules, we construct the expert's view that consists of all the terms used in the rules. After that, the expert gives ten new concept names, for which the system as well as the expert construct the rules independently. Using the both kinds of rules, we retrieve documents. Actually, the expert creates five OR rules and five AND rules for the given ten concepts.

5.1. OR rules

We first consider the OR rules. The five OR rules are as follows.

**Expert's OR rules**

1. leaves | mast | duff => litter
2. fragmentation | forest edges => ecotones
Table 1. The result for the OR rules made by the expert

<table>
<thead>
<tr>
<th>concept</th>
<th>No. of Retrieved Documents</th>
<th>No. of Relevant Documents</th>
<th>No. of Possibly Relevant Documents</th>
<th>No. of Irrelevant Documents</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>507</td>
<td>146</td>
<td>56</td>
<td>305</td>
<td>0.40</td>
</tr>
<tr>
<td>(2)</td>
<td>134</td>
<td>26</td>
<td>24</td>
<td>84</td>
<td>0.37</td>
</tr>
<tr>
<td>(3)</td>
<td>273</td>
<td>26</td>
<td>16</td>
<td>231</td>
<td>0.15</td>
</tr>
<tr>
<td>(4)</td>
<td>590</td>
<td>200</td>
<td>89</td>
<td>301</td>
<td>0.49</td>
</tr>
<tr>
<td>(5)</td>
<td>349</td>
<td>67</td>
<td>40</td>
<td>242</td>
<td>0.31</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.34</td>
</tr>
</tbody>
</table>

Table 2. The result for the generated rules based on Type 3 rule

<table>
<thead>
<tr>
<th>concept</th>
<th>No. of Retrieved Documents</th>
<th>No. of Relevant Documents</th>
<th>No. of Possibly Relevant Documents</th>
<th>No. of Irrelevant Documents</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>394</td>
<td>156</td>
<td>60</td>
<td>178</td>
<td>0.55</td>
</tr>
<tr>
<td>(2)</td>
<td>192</td>
<td>36</td>
<td>37</td>
<td>119</td>
<td>0.38</td>
</tr>
<tr>
<td>(3)</td>
<td>102</td>
<td>53</td>
<td>21</td>
<td>28</td>
<td>0.73</td>
</tr>
<tr>
<td>(4)</td>
<td>249</td>
<td>46</td>
<td>34</td>
<td>169</td>
<td>0.32</td>
</tr>
<tr>
<td>(5)</td>
<td>14</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>0.71</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.54</td>
</tr>
</tbody>
</table>

Table 3. The result for the generated rules based on Type 1 rule

<table>
<thead>
<tr>
<th>concept</th>
<th>No. of Retrieved Documents</th>
<th>No. of Relevant Documents</th>
<th>No. of Possibly Relevant Documents</th>
<th>No. of Irrelevant Documents</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>(2)</td>
<td>192</td>
<td>36</td>
<td>37</td>
<td>119</td>
<td>0.38</td>
</tr>
<tr>
<td>(3)</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.67</td>
</tr>
<tr>
<td>(4)</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.50</td>
</tr>
<tr>
<td>(5)</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0.71</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.59</td>
</tr>
</tbody>
</table>

(3) carnivorous plants | organic soils | humus
=> bogs

(4) marshlands | fens | bogs | marshes | swamps
=> wetlands

(5) dust | ash | soot
=> particulate

The right hand sides of the above rules are the given concept names. The left hand sides of the rules are constructed by the expert. Using the left hand sides, we retrieve documents. For the retrieved documents, the expert decides whether they are relevant, possibly relevant, or irrelevant. Table 1 shows the result. The precision is computed as (the number of relevant documents + the number of possibly relevant documents) / the number of retrieved documents. Recall is not considered because of the size of the database.

For the above five concepts, our system automatically generates rules, by using the function find, in the following two ways. The first way is to use Type 3 rule (OR rule). Using Type 3 rule, our system generates the rules are as in the below. To generate the rules, we use the following parameters: the weighting value for the relation NT is 0.8, for BT 0.7, for RT 0.6; the maximum depth level of activation is 6; and the threshold for the activation
weighting value is 0.2. To construct the left hand sides of the rules, the system chooses the first three terms having the highest weighting values computed from the spreading activation. The precision rates are shown in Table 2. The average precision is higher than that in Table 1.

System’s Type 3 rules (OR rules)

(1) humus | fuel types => litter
(2) forest fragmentation => ecotones
(3) marshlands => bogs
(4) riparian habitats | bogs => wetlands
(5) dust | soot => particulate

The second way for generating the rules is to use Type 1 rule. The generated rules are shown in the below, and the resulting precision rates are shown in Table 3. We use the same parameters as we used for the first way. Although the result of using Type 1 rule is different from the result of using the expert’s OR rules in the amount of retrieved documents, the resulting precision rates are much better than those in Table 1.

System’s Type 1 rules

(1) humus & fuel types & fine fuels => litter
(2) forest fragmentation => ecotones
(3) savannas & marshlands => bogs
(4) riparian habitats & bogs & grasslands => wetlands
(5) dust | soot & air quality => particulate

5.2. AND rules

We consider the expert’s AND rules. For the AND rules, actually two experts (expert A and expert B) are involved. They construct two five rules independently. First, let us see the expert A’s five AND rules in the below. The resulting precision rates are shown in Table 4.

Expert A’s AND rules

(1) water fowl & hunting => wildlife management
(2) mast & seeds => fruits
(3) marshlands & swamps => riparian habitats
(4) smoke & fire control => smoke management
(5) forest fragmentation & site treatments

System’s Type 2 Rules for Expert A’s AND Rules

(1) birds & wildlife & wildlife food habits => wildlife management
(2) mast & seeds => fruits
(3) habitat types & streams & wildlife habitat management => riparian habitats
(4) smoke & air quality & convection => smoke management
(5) fire control & fuel breaks => firebreaks

Now, let us consider the expert B’s five AND rules shown as follows. The resulting precision rates are given in Table 6.

Expert B’s AND rules

(1) fuels & weather & flame length => fire behavior
(2) temperature & flame length => fire intensity
(3) urban wildland fire interface & rural communities => fire prevention
(4) wetlands & bottomland hardwoods => fire frequency
(5) logging & fragmentation => wildfires

For the above concepts, our system generates the rules as follows, and the result is shown in Table 7. Our precision rates are better than those in Table 6, even if we ignore concept (5) in Table 7 because of the small number of retrieved documents.

System’s Type 2 Rules for Expert B’s AND Rules

(1) fire weather & fire case histories & fire intensity => fire behavior
(2) fire regimes & catastrophic fires & crown fires => fire intensity
(3) fire control & fire hazard reduction & public information => fire prevention
(4) burning intervals & fire exclusion & fire history => fire frequency
(5) prescribed fires & disturbance & grass fires => wildfires

3 From the predefined rules, we can statistically find that the expert uses three terms for constructing the rules, and we eliminate the terms having more than 350 retrieved documents since the database allows to display up to 350 documents.
Table 4. The result for the AND rules made by expert A

<table>
<thead>
<tr>
<th>concept</th>
<th>No. of Retrieved Documents</th>
<th>No. of Relevant Documents</th>
<th>No. of Possibly Relevant Documents</th>
<th>No. of Irrelevant Documents</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>13</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>0.62</td>
</tr>
<tr>
<td>(2)</td>
<td>12</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>0.83</td>
</tr>
<tr>
<td>(3)</td>
<td>23</td>
<td>5</td>
<td>5</td>
<td>13</td>
<td>0.43</td>
</tr>
<tr>
<td>(4)</td>
<td>34</td>
<td>9</td>
<td>10</td>
<td>15</td>
<td>0.56</td>
</tr>
<tr>
<td>(5)</td>
<td>22</td>
<td>7</td>
<td>4</td>
<td>11</td>
<td>0.50</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.59</td>
</tr>
</tbody>
</table>

Table 5. The result for the generated rules based on Type 2 rule (Expert A)

<table>
<thead>
<tr>
<th>concept</th>
<th>No. of Retrieved Documents</th>
<th>No. of Relevant Documents</th>
<th>No. of Possibly Relevant Documents</th>
<th>No. of Irrelevant Documents</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>36</td>
<td>19</td>
<td>10</td>
<td>7</td>
<td>0.81</td>
</tr>
<tr>
<td>(2)</td>
<td>12</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>0.83</td>
</tr>
<tr>
<td>(3)</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>(4)</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>(5)</td>
<td>17</td>
<td>6</td>
<td>8</td>
<td>3</td>
<td>0.82</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.89</td>
</tr>
</tbody>
</table>

Table 6. The result for the AND rules made by expert B

<table>
<thead>
<tr>
<th>concept</th>
<th>No. of Retrieved Documents</th>
<th>No. of Relevant Documents</th>
<th>No. of Possibly Relevant Documents</th>
<th>No. of Irrelevant Documents</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>0.58</td>
</tr>
<tr>
<td>(2)</td>
<td>15</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>0.60</td>
</tr>
<tr>
<td>(3)</td>
<td>32</td>
<td>17</td>
<td>6</td>
<td>9</td>
<td>0.72</td>
</tr>
<tr>
<td>(4)</td>
<td>28</td>
<td>3</td>
<td>10</td>
<td>15</td>
<td>0.46</td>
</tr>
<tr>
<td>(5)</td>
<td>11</td>
<td>0</td>
<td>7</td>
<td>4</td>
<td>0.64</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 7. The result for the generated rules based on Type 2 rule (Expert B)

<table>
<thead>
<tr>
<th>concept</th>
<th>No. of Retrieved Documents</th>
<th>No. of Relevant Documents</th>
<th>No. of Possibly Relevant Documents</th>
<th>No. of Irrelevant Documents</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>26</td>
<td>17</td>
<td>7</td>
<td>2</td>
<td>0.92</td>
</tr>
<tr>
<td>(2)</td>
<td>30</td>
<td>1</td>
<td>18</td>
<td>11</td>
<td>0.63</td>
</tr>
<tr>
<td>(3)</td>
<td>12</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>(4)</td>
<td>28</td>
<td>9</td>
<td>11</td>
<td>8</td>
<td>0.71</td>
</tr>
<tr>
<td>(5)</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.85</td>
</tr>
</tbody>
</table>
6. Conclusions

In this paper, we clarify the semantics for the rules used in RUBRIC that can be used to describe queries as a conceptual level, and propose a way to construct concepts from thesauri based on the semantics. The generated rules are more effective than hand-made rules in terms of precision. However, in order to generate rules, we used the prefixed weighting values for the relationships NT (Narrow Term), BT (Broad Term), and RT (Related Term), since thesauri usually do not provide degrees of relatedness (or closeness) between terms. Therefore, whenever we use a different thesaurus, we need to adjust the weighting values for the relationships. In the future, we need to do more experimental tests on various thesauri and to propose better ways to generate user's view that can reflect user's interest.

Acknowledgements

We would like to thank two domain experts, Judy Buys and Dr. Susan Grace in National Wetlands Resource Center at Lafayette Louisiana, for constructing the rules and deciding the relevancy on the retrieved documents.

References


