Automatic Text Categorization based on Hierarchical Rules

Minoru SASAKI and Kenji KITA
Faculty of Engineering, Tokushima University
Tokushima 770-8506, JAPAN

Key Words: document categorization, rule learning, hierarchical category, RIPPER

Abstract

Document categorization, which is defined as the classification of text documents into one of several fixed classes or categories, has become important with the explosive growth of the World Wide Web. The goal of the work described here is to automatically categorize Web documents in order to enable effective retrieval of Web information. In this paper, based on the rule learning algorithm RIPPER (for Repeated Incremental Pruning to Produce Error Reduction), we propose an efficient method for hierarchical document categorization.

1 Introduction

Recently, as the World Wide Web(WWW or Web) developed rapidly, a large collection of full-text documents in electronic form is available and opportunity for getting a useful piece of information is increased. Also in the WWW it is quite common to have large, manually ordered collections of hypertext links (e.g. Yahoo) and it is effective to refer to the links. Text categorization is the classification of texts with respect to a set of categories that are predefined. Traditionally, its task has been done by human experts. But as the number of texts increases, this task becomes more difficult for human experts. Under the circumstances, this has led to increased interest in automatic methods for filtering and categorizing documents.

There are many approaches to text categorization, such as rule-based, knowledge-based, text-similarity-based and so on. We focus on the approach that learn sets of rules from given training documents. Rule representation is relatively easy for people to understand and prior knowledge can be easily communicated to other rule learning systems.

In this paper, we point out the problem of automatic document categorization by using the rule learning algorithm RIPPER. We extend the algorithm to construct a set of hierarchical rules automatically besides a set of rules RIPPER constructs.

2 Rule-Based Text Categorization

Many methods for text categorization have been studied with the aim of efficient extraction of useful information from a huge resource. From these methods, we direct our attention to the rule learning approach.

RIPPER rule learning algorithm, an extended version of learning algorithm IREP(Incremental Reduced Error Pruning)[1][2], constructs a ruleset that all positive examples are covered, and its algorithm perform efficiently on large, noisy datasets. This algorithm is described in detail by [3][4], but we will summarize it below. Before building a rule, the current set of training examples is partition into two subsets, a growing set(usually 2/3) and a pruning set(usually 1/3). The rule is constructed from examples in the growing set. And then, the ruleset begins with an empty ruleset and rules are added incrementally to the ruleset until no negative examples are covered.

After growing a rule from the growing set, condition is deleted from the rule in order to improve the performance of the ruleset on the pruning examples. To prune a rule, RIPPER considers only a final sequence of conditions from the rule, and selects the deletion that maximizes the function

$$v(Rule, PrPos, PrNeg) \equiv \frac{p-n}{p+n} \qquad (1)$$

where $Rule$ is the set of rules, $PrPos$ is the total number of examples in the considered cluster, $PrNeg$ is the total number of examples in the cluster not considered

and p (n) is the number of $PrPos$ ($PrNeg$) examples covered by $Rule$. Whenever no deletion improves the value of function v, learning stops. Furthermore, after the rule is added to the ruleset, the total description length of the rule is computed. When the longest description length is more than 64 bits larger than the smallest one, learning also stops. All covered positive and negative examples are removed from growing and pruning set and a new rule is constructed from the remaining examples.

An example of a ruleset RIPPER constructs is as follows(using Prolog-like notation).

Painting :- WORDS~"watercolor".

Painting :- WORDS~"art", WORDS~"museum".

Painting :- WORDS~"author", WORDS~"picture".

This ruleset means that a document d is considered to be in the category "Painting" if and only if

(word "Watercolor" appears in d) OR
(word "art" in d AND word "museum" in d) OR
(word "author" in d AND word "picture" in d).

3 Hierarchical Document Categories

One of the problems with the RIPPER algorithm is that it deletes a condition for a word which appears in two or more categories. As an simple example, we consider two similar categories "Photography" and "Painting" and construct a ruleset from the training data which belongs to these categories. In this data, the word "gallery"may appear frequently in categories "Photography" and "Painting". To achieve the accurate categorization, the word "gallery" is deleted. Thus, RIPPER deletes the condition for the word "gallery" in the pruning phase and does not construct the following rules:

Painting :- WORDS~"gallery".

Photography :- WORDS~"gallery".

The rule is deleted to improve the performance of the ruleset on the pruning set. When no rules cover a test data, the test data is categorized into the category that the maximum number of texts belong to in the training data and this category will be used as the defualt category. So the number of conditions in the ruleset decreases and conditions in the ruleset may not appear in the test data. In order to decrease a failure

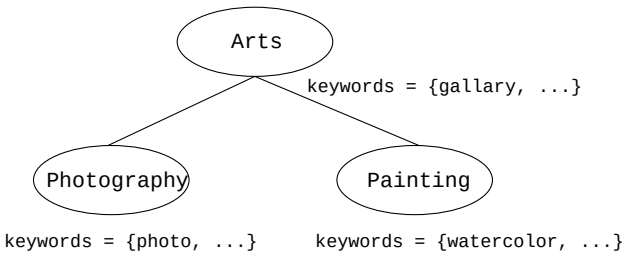

Figure 1: Structure of hierarchical category

of retrieval, the possibility of retrieval may increase by adding some condition that correspond to words to the ruleset.

To avoid the problem, we extended the RIPPER algorithm to introduce hierarchical categories in a ruleset automatically. hierarchical categories means that a new category which covers both categories is constructed. When a word appears frequently in two or more categories, a new category will be constructed. For example, we create rules shown below:

Arts :- CATEGORY~Painting.

Arts :- CATEGORY~Photography.

Arts :- WORDS~"gallery".

Painting :- WORDS~"watercolor".

The new category "Arts" covers both of the category "Painting" and "Photography", and a document d is considered to be in the category "Arts" if the word "gallery" appears in d. (We use the category name "Arts" for the sake of convenience. In practice, category names are automatically generated by a program.) This hierarchical ruleset is illustrated in Figure 1.

Figure 2 presents the extended RIPPER algorithm. The original RIPPER algorithm removes all covered positive and negative examples from the training data. The removed positive examples are used to select words for a rule and the removed negative examples are used to refine their words. Then, the selected word is deleted because the same word appears in other categories. Finally, the word which appears only in one category is considered as a condition of a rule. The other words are not used in the RIPPER algorithm after the refinement of the condition of the rule.

In the extended algorithm, their unselected words are used to make the hierarchical categories and a rule by which all examples in both of two categories are covered. For any category, all unselected words are

```
procedure HierarchicalRIPPER(Pos, Neg)
begin
  Ruleset := 0
  while Pos ≠ 0 do
    split(Pos, Neg) into (GrowPos, GrowNeg)
      and (PrunePos, PruneNeg)
    GrowRule := Grow(GrowPos, GrowNeg)
    PruneRule := Grow(PrunePos, PruneNeg)
    if |GrowRule| − |PruneRule| > threshold
    then
      return Ruleset
    else
      add Rule to Ruleset remove examples
        covered by Rule from (Pos, Neg)
      if Removed example has higher
        score than threshold score
      then
        add Rule as a class covered by
          both classes to Ruleset
      endif
    endif
  endwhile
  return
end
```

Figure 2: Hierarchical RIPPER algorithm

extracted and thay are examined whether the same word appear in both of categories. If the frequency of the same word is both higher than a certain constant value in the two categories, a new category which covers both categories and a hierarchical rule which represents this word belongs to the both categories is constructed.

4 Experimental Results

We have conducted comparative experiments using the original RIPPER algorithm and the hierarchical RIPPER algorithm. Japanese Web pages linked from *Yahoo! Japan*(http://www.yahoo.co.jp/) were collected and morphologically analized to extract noun words or phrases because Japanese texts have no word delimiter between words and are used as the training data. The remaining words such as adjective and verb and so on are regarded as language-specific function words, so their words are not treated. In our experiment, the collected Web pages, a total of 1979 documents, were divided into two sets, 1832 training documents and 147 testing documents.

The results are summarized in confusion matrices and they are presented in Table 1 (the original RIPPER algorithm) and Table 2 (our hierarchical RIPPER algorithm). Category names (A, B, $\cdots$, J) correspond to the Yahoo's classification as follows:

A Arts/Humanities

B Arts/Visual_Arts

C Computers_and_Internet/Internet /World_Wide_Web

D Computers_and_Internet /Programming_Languages

E Computers_and_Internet/Operating_Systems

F Computers_and_Internet/Hardware /Personal_Computers

G Computers_and_Internet/Software

H Entertainment/Movies_and_Films

I Entertainment/Music

J Computers_and_Internet/Graphics

In Table 2, "NewCat" means that documents are categorized in the new category created automatically by the hierarchical RIPPER algorithm.

As can be seen from Table 1, the RIPPER tends to miscategorize documents into category "C" (Computers_and_Internet/Internet/World_Wide_Web). If the number of conditions in the ruleset is small by the deletion of conditions or rules, It is hard to categorize a test data into a true category. If a test data is not covered by any rule in the ruleset, the test data categorizes into the default category which has the maximum number of documents. But the hierarchical RIPPER does not tend to miscategorize their documents into the default category, as can be seen from Table 2. Consequently, we can say that the hierarchical RIPPER is superior to the original RIPPER.

5 Conclusions

Motivated by an increased interest in automatically categorizing the World Wide Web documents and improving the performance of the RIPPER rule learning algorithm, in this paper we proposed a new algorithm for document categorization based on the RIPPER algorithm. We have obtained encouraging results. The reason that our algorithm is superior to the original algorithm is that our algorithm uses hierarchical categories to increase the number of words in the ruleset.

Input	Category Identified									
	A	B	C	D	E	F	G	H	I	J
A	16	1	9	0	0	0	0	0	0	0
B	0	4	4	0	0	0	0	0	0	0
C	1	1	20	1	1	0	0	2	0	0
D	0	0	3	0	0	0	0	0	0	0
E	0	0	17	0	8	0	0	0	0	0
F	0	0	5	0	0	0	0	0	0	0
G	0	0	15	1	2	0	1	0	1	0
H	0	0	6	1	0	0	1	10	0	0
I	0	0	8	0	0	0	0	0	5	0
J	0	0	3	0	0	0	0	0	0	0

Table 1: Result obtained by the original RIPPER algorithm

Input	Category Identified										
	A	B	C	D	E	F	G	H	I	J	NewCat
A	16	1	1	0	0	0	0	0	0	0	8
B	0	4	0	0	0	0	0	0	0	0	5
C	1	1	17	1	1	0	0	2	0	0	3
D	0	0	0	0	0	0	0	0	0	0	3
E	0	0	1	0	8	0	0	0	0	0	16
F	0	0	0	0	0	0	0	0	0	0	5
G	0	0	2	1	2	0	1	0	1	0	13
H	1	0	3	0	0	0	0	10	0	0	3
I	0	0	6	0	0	0	0	0	5	0	6
J	0	0	0	0	0	0	0	0	0	0	3

Table 2: Result obtained by the hierarchical RIPPER algorithm

As future research, we intend to elaborate the method by combining different categorization methods such as probabilistic classifiers.

References

[1] Johannes Fürnkranz, Gerhard Widmer: "Incremental Reduced Error Pruning", In W. Cohen and H. Hirsh, editors, Proceedings of the 11th International Conference on Machine Learning (ML-94), pp. 70-77, New Brunswick, NJ, 1994. Morgan Kaufmann.

[2] Johannes Fürnkranz: "A Tight Integration of Pruning and Learning", In N. Lavrac and S. Wrobel, editors, Proceedings of the 8th European Conference on Machine Learning (ECML-95), pp. 291-294, Crete, Greece, 1995. Springer-Verlag.

[3] William W. Cohen: "Fast Effective Rule Induction", Proceedings of the Twelfth International Conference on Machine Learning, Lake Tahoe, California, 1995.

[4] William W. Cohen: "Learning to Classify English Text with ILP Methods", In Advances in Inductive Logic Programming (Ed. L. De Raedt), IOS Press, 1995.