Rule Induction for Concept Hierarchy Alignment

ICHISE Ryutaro, TAKEDA Hideaki. and HONIDEN Shinichi
Intelligent Systems Research Division,
National Institute of Informatics,
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo, 101-8430, Japan
{ichise,takeda,honiden}@nii.ac.jp

Abstract

To manage information like ontology, we usually use categorization with concept hierarchy. Such concept hierarchies are managed individual for each system due to the many differences in concept hierarchies. Consequently, it is difficult to reuse information in computer-based systems. Here, we propose a new concept alignment method for concept hierarchies as a solution to this problem, and construct a system to evaluate the performance of our method. The results of this experiment reveal that the proposed method can be used to induce appropriate align rules for concept hierarchies and classify information into appropriate categories within another concept hierarchy.

1 Introduction

The rapid advances in computer technology have allowed us to archive much more information than ever before. To maintain such archives, hierarchical categorization is often used, by which information is categorized based on a concept hierarchy. Ontologies and class libraries are examples of such systems. The hierarchical categorization is usually maintained by a single organization or individual for consistency. In practice, a concept hierarchy is not suitable for multiple purposes, requiring that concept hierarchies be constructed for individual application domains. In other words, concept hierarchies are maintained in each domain separately. This means that it is not convenient to reuse information because the concept categorization differs between concept hierarchies. To solve this problem, we propose a new method that allows a concept in one concept hierarchy to be aligned with another concept in another concept hierarchy. In this paper, we describe a machine-learning method for aligning multiple concept hierarchies.

This paper is organized as follows: A concept hierarchy model is defined, followed by the proposal for the new machine learning method used to align concepts. In the experiment section, the performance of our system, HICAL (HIerarchical Concept ALignment system), which is based on the proposed method, is compared in a variety of settings, followed by a discussion of the results. We then discuss related work in regard to HICAL and present the conclusions of this study.

2 Concept Hierarchy Model

In this section, we describe a model of the nature of concept hierarchies. Many information management systems for use with conceptual information like ontologies and class libraries are managed via a system of hierarchical categorization. A concept hierarchy contains only one conceptual categorization criterion. We can consider the diagram on the left side of Figure 1 to be a representation of a single concept hierarchy.

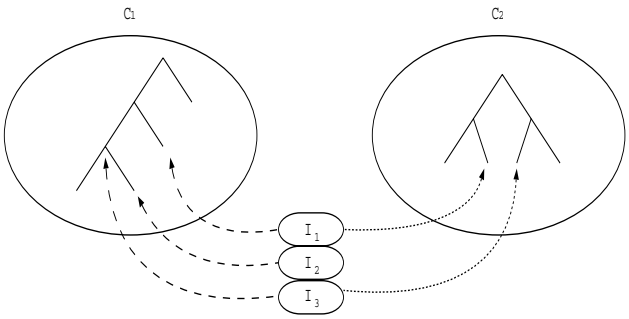

Figure 1: Concept Hierarchy Model

In the figure there are two different concept hierarchies (C_1 and C_2) and three different information instances (I_1, I_2 and I_3). Some instances are shared between the two concept hierarchies and some are not. It is important to keep in mind that these concept hierarchies are not the same. The next step is to consider an appropriate way to transfer an instance from C_1 into C_2. In the example shown in Figure 1, C_2 does not contain I_2. If I_2 can be placed in the concept hierarchy of C_2, the user can then use I_2 with concept hierarchy C_2. In the next section, we propose a method of defining a rule for conversion from the concept hierarchy of C_1 to that of C_2, so that an instance that has already been categorized in C_1 can subsequently be categorized in C_2. The most important point of this approach is that the concept hierarchy of C_2 does not need to be adjusted to fit the concept hierarchy of C_1. Thus, a

		Category S_1	
		belong	not belong
Category S_2	belong	N_{11}	N_{12}
	not belong	N_{21}	N_{22}

Table 1: Classification of Instances by Two Categories

user can apply our method while continuing to use whichever concept hierarchy they are accustomed to.

3 Concept Alignment

In our proposed method of constructing an alignment rule, we must first find categories that are similar to each other ("similar categories"); the instance will then be aligned based on the similarity between categories. To find similar categories, our algorithm starts by comparing the most general categories of the two concept hierarchies. For each pair of categories, we can determine similarity based on the instances categorized in the two categories. For each category, we can decide whether a particular instance belongs to that category. If the categorization methods of two categories are similar, then the system can generate an aligning rule for them. For example, if one category contains 100 instances and a category in another concept hierarchy contains the same 100 instances, then the algorithm can generate an aligning rule for these categories, because they can be considered to have the same categorization criteria. It should be noted that, because concept hierarchies are structured as trees, we can easily categorize according to a nodal structure, such that lower (more specific) categories are included in higher (more general) categories.

To find similar categories, we used a statistical method for determining the degree of similarity between two categorization criteria. The "κ statistic" method [Fleiss, 1973] is an established method of evaluating similarity between two criteria. We explain this method briefly. Let us suppose that there are two categorization criteria, S_1 and S_2. As mentioned earlier, we can decide whether a particular information instance belongs to a particular category or not. Consequently, instances are divided into four classes shown in Table 1. Symbols $N_{11}, N_{12}, N_{21}, N_{22}$ denote numbers of instances for each class. For example, N_{11} denotes the number of instances which belong to both the category S_1 and the category S_2. We may logically suppose that if category S_1 and S_2 have the same criterion of categorization, then N_{12} and N_{21} become close to zero and if the two categories have a different criterion of categorization, then N_{11} and N_{22} become close to zero. The "κ statistic" method utilizes this principle to determine the similarity of categorization criteria.

The relationship between the two categorization criteria is examined from "top" to "bottom". The alignment algorithm is shown in Figure 2. First, the most general categories in the two concept hierarchies are compared using the "κ statistic". If the comparison confirms that the two categories are similar, then the algorithm outputs an alignment rule for them. At the same time, the algorithm pairs one of these two similar categories with a "child" category in the other similar category. This new pair is then evaluated recursively using the "κ statistic" method. When a similar pair is not generated, the algorithm outputs the alignment rule between the two concept hierarchies. We can then apply this rule to deciding whether a particular instance in C_1 fits the concept hierarchy in C_2.

```
Input:   N10,  // Top category in C1
         N20,  // Top category in C2
         P;    // threshold for κ statistic
Output:  R;    // Rule Set
begin
  /* make pair for candidate */
  /*  using child node or parent node */
  X1 := make_combination(N10, N20);
  t := 1;
  R := φ;
  while Xt ≠ φ
    while Xt ≠ φ
      I := element in Xt;
      N1, N2 := two node in I;
      /* calculate κ statistic */
      if κ(N1, N2) ≥ P
        Xt+1 := make_combination(N1, N2);
        R := R + I;
      fi;
      Xt := Xt − I;
    end;
    t := t + 1;
  end;
  return R;
end;
```

Figure 2: Alignment Algorithm

4 Experimental Evaluation

4.1 Data and Settings

In order to evaluate this algorithm, we conducted three experiments using the Yahoo! Japan [Yahoo! Japan, 2000] and LYCOS Japan [LYCOS Japan, 2000] directories as concept hierarchies, and the links (URLs) in each directory as information instances. The Yahoo! directory contains approximately 41,000 categories and 224,000 URLs. LYCOS contains approximately 5,700 categories and 48,000 URLs. Approximately 25,000 URLs are common to both Yahoo! and LYCOS. Generally speaking, as a concept hierarchy, Yahoo! contains more knowledge than LYCOS, however half of the URLs in LYCOS are not contained in Yahoo!. This demonstrates that even a concept hierarchy that contains an enormous amount of information does not cover all information.

In this study, we used the three category pairs (and subcategories) as our two concept hierarchies. The location in Yahoo! and LYCOS are as follows:

- Yahoo! : Arts / Humanities / Literature
  LYCOS : Arts / Literature

- Yahoo! : Business and Economy / Companies
  LYCOS : Business Industry / Company
- Yahoo! : Recreation
  LYCOS : Hobby Sports

We conducted 10-fold cross validation for shared instances. Shared instances were divided into 10 data sets; 9 of these sets were used for training and the remaining set was used for testing. Ten experiments were conducted for each data set. The parameter of significance level for the "κ statistic" was set at 5%.

4.2 Results

The results of the experiments are shown in Figure 3, 4 and 5. The vertical axis is the average accuracy of the test data. "Exact rules" represent values of accuracy for a system that only uses the alignment rules for the category to which the instance belongs. "Parent rules" indicates that if the system does not generate an alignment rule for a category to which the instance belongs, it will use the rule generated for the parent category instead. "Criterion 1" indicates that the instance is categorized in the same category as the test data and "Criterion 2" indicates that the instance is categorized in the same category or parent category as the test data. "Criterion 1" is very strict criterion because the target concept hierarchy should have enough intermediate categories in comparison with the source concept hierarchy, while "Criterion 2" is more general and more realistic because it does not matter which concept hierarchies are rich in categorization. Two directions of alignment are presented; from Yahoo! to Lycos, and from Lycos to Yahoo!.

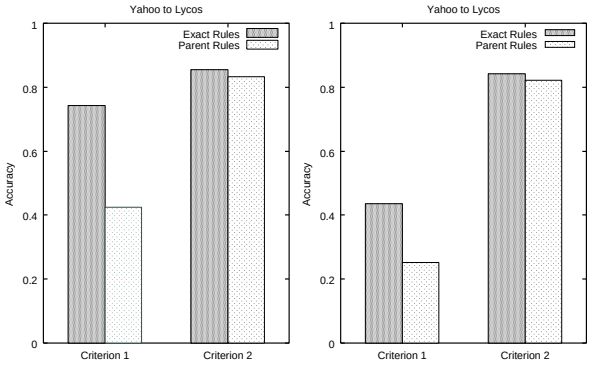

Figure 3: Result for Literature Domain

4.3 Discussion

More than 80% of the instances used in our experiments, with the exception of the company domain, were categorized correctly by HICAL. In the company domain, more than 60% of the instances were categorize appropriately. The data in Figure 3$\sim$ 5 imply that Yahoo-to-Lycos alignment was more accurate than the inverse operation. As can be seen from the total number of categories, the categorical hierarchy in Yahoo! is more complex than that of LYCOS. Therefore, for Yahoo-to-Lycos alignment, the learned rules are likely to

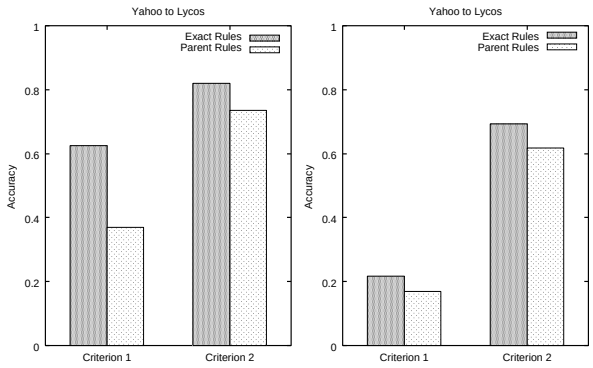

Figure 4: Result for Company Domain

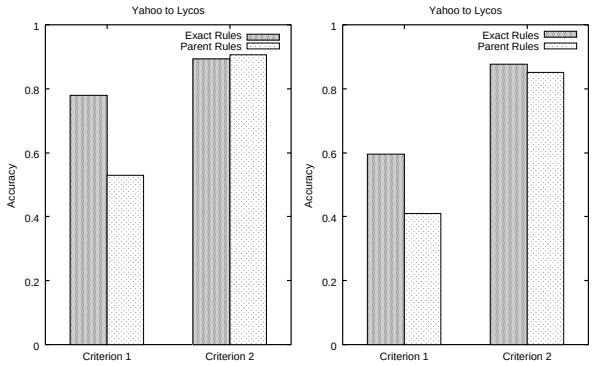

Figure 5: Result for Recreation Domain

involve transferring instances from relatively complex categories to relatively general categories. Such a trend is likely to result in relatively accurate rules. For example, suppose concept hierarchy A contains category S within concept X, however concept hierarchy B does not divide X into categories. In such a case, it would be much easier to learn a rule for "A:/X/S -> B:/X" than to learn a rule for "B:/X -> A:/X/S", because "B:/X" contains instances that belong in S and instances that do not. The results shown in figures reflect this. Nevertheless, in this situation our method works properly. When regarding parent categories as correct answers ("Criterion 2"), both alignment directions exhibited almost the same results, i.e., "B:/X -> A:/X" is learned instead of "B:/X -> A:/X/S".

The limitation of this method is that it does not work in cases in which the more general and more specific categories within a concept hierarchy are independent. For example, let's consider two concept hierarchies; one that classifies a food-related instance under foodstuff as the more general category and then by country of origin as the more specific category, and another that classifies a food-related instance by country of origin as the more general category and by foodstuff as the more specific category. Both types of concept hierarchy can be found on the concept hierarchy. In such a case, our current system HICAL would not work, because comparison between the two categories proceeds from gen-

eral to specific (top to bottom). To solve this problem, we are planning to combine our current system with a bottom-up method.

5 Related Work

One of the systems related to HICAL is the ontology merging/alignment system. In the merging process for ontology, a process such as our system is necessary due the requirements of concept hierarchy management. Chimaera [McGuinness *et al.*, 2000] and PROMPT [Noy and Musen, 2000] are examples of such systems, assisting in the combination of different ontologies. However, such systems require human interaction for merging or alignment. In addition to this requirement, they are based on similarity between words, which introduces instability. Word similarity is often biased by the dictionaries used. In contrast, our system does not use word similarity, instead using syntactics alone. Hence, our system has the ability to find identical concepts regardless of the category name or word. For example in the experiment conducted in this study, in the literature domain, HICAL found the relationship between the "Genji-monogatari" (a famous Japanese story written by "Murasakishikibu") category in LYCOS and the "Murasakishikibu" category in Yahoo![1]. In LYCOS, classical literature is classified by title (concept category), whereas in Yahoo!, poetry masters are categorized by author. As the dictionaries commonly used do not contain such information, word-based systems would not be capable of finding title/author relationships.

The bookmark-sharing systems of Siteseer [Rucker, 1997] and Blink [Blink, 2000] are also similar to HICAL. The main difference is the use of hierarchies for categorization. The Siteseer and Blink system only considers the number of URLs (instances) in a given category, whereas our method uses hierarchical structures. One of the merits of our approach is that if there is no exact category into which a given URL (instance) fits, then the URL (instance) is mapped into the parent category. kMedia [Takeda *et al.*2000] is another bookmark-sharing system that uses hierarchical structures explicitly but is dependent on similarity of words in pages. Bookmark-Agent [Mori and Yamada, 1999] uses another approach, utilizing bookmarks based on keywords. As mentioned above, HICAL only uses syntactical information, not words as are used by a bookmark agent. HICAL is therefore capable of correctly categorizing different words under the same concept.

6 Conclusion

In this paper, we propose a new method for aligning concept hierarchies as a new approach to utilizing information in multiple concept hierarchies, based on statistical methods. To test our ideas, we conducted experiments using the Yahoo! and LYCOS categories. Our experimental results show that the alignment rules learned by HICAL yield reliable alignments, allowing information in one concept hierarchy to be aligned to an appropriate position in another concept hierarchy. The advantage of using our method is that it allows users to use their own concept hierarchy for categorizing all information, and may serve as a powerful tool for aligning concept hierarchies.

With these encouraging results, several research possibilities present themselves for future development of alignment strategies. Our alignment method is based on a top-down approach. We should combine a bottom-up approach to increase accuracy. In addition, there may exist other possibilities for alignment. Extending the proposal to applying to more than three concept hierarchies needs to be investigated. In such a case, despite confliction between several concept hierarchies, more hints can be obtained from other concept hierarchies.

[1] similar to the relationship between "Sherlock Holmes" and "Conan Doyle"

References

[Blink, 2000] Blink. *http://www.blink.com/*, 2000.

[Fleiss, 1973] Joseph L. Fleiss. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, 1973.

[LYCOS Japan, 2000] LYCOS Japan. *http://www.lycos.co.jp/*, 2000.

[McGuinness *et al.*, 2000] Deborah L. McGuinness, Richard Fikes, James Rice and Stive Wilder. An Environment for Merging and Testing Large Ontologies In *Proceedings of the seventh International Conference on Principles of Knowledge Representation and Reasoning(KR2000)*, Morgan Kaufman Publishers, 2000.

[Mori and Yamada, 1999] Mikihiko Mori and Seiji Yamada. Bookmark-Agent: Information Sharing of URLs In *Poster Proceedings of the 8th International World Wide Web Conference(WWW-10)*, 1999.

[Noy and Musen, 2000] Natalya F. Noy and Mark A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 450–455, Austin, Texas, July–August 2000. American Association for Artificial Intelligence.

[Rucker, 1997] James Rucker and Marcos J. Polanco. Siteseer: Personalized Navigation for the Web *Communications of the ACM*, 40(3):73–75, 1997.

[Takeda *et al.*2000] Hideaki Takeda, Takeshi Matsuzuka and Yuichiro Taniguchi. Discovery of Shared Topics Networks among People - A Simple Approach to Find Community Knowledge from WWW Bookmarks In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence(PRICAI-2000)*, pages 668–678, Melbourne, Australia, August 28 - September 1, Springer, 2000.

[Yahoo! Japan, 2000] Yahoo! Japan. *http://www.yahoo.co.jp/*, 2000.