ISSN 1346-5597

NII National Institute of Informatics

NII Technical Report

An Alignment Algorithm between Concept Hierarchies

ICHISE Ryutaro, TAKEDA Hideaki, and HONIDEN Shinichi

NII-2001-001E
May 2001

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

An Alignment Algorithm between Concept Hierarchies

Ichise, R., Takeda, H. and Honiden, S.
Intelligent Systems Research Division,
National Institute of Informatics,
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo, 101-8430, Japan
{ichise,takeda,honiden}@nii.ac.jp

May 10, 2001

Abstract

Hierarchical categorization is a powerful and convenient method so that it is commonly used in various areas, for example ontologies and information categorization. Although each hierarchy is useful, there are problems to manage multiple hierarchies. In this paper, we propose an alignment method between concept hierarchies by using a statistical method. By using this method, a concept that exists in one hierarchy system but does not in the other can be located in a suitable position in the other. The key idea is to find similar categories between two systems to be able to transfer concepts from one system to the other. Similarity is measured by "κ(kappa) statistic" based on instances belonging categories. The experiments of our method with concept hierarchies of Yahoo! and LYCOS result over 80% of accuracy to estimate appropriate positions of concepts between two hierarchies.

1 Introduction

The rapid advances in computer technology have allowed us to archive much more information than ever before. Categorization is one of the important issues to manage such archive so that the importance of ontology as a conceptual system is recently focused [Guarino *et al.*1995]. Usually ontologies are organized as a hierarchical structure. Hierarchical categorization is a powerful and convenient method so that it is commonly used in various areas. Although each hierarchy is useful, there are problems to manage multiple hierarchies. There are two possibilities to manage multiple hierarchies, i.e., merging and aligning [Noy *et al.*1999]. Merging ontologies is preferred if these ontologies are consistent totally. But this situation is not so common because each ontology has its aspect for categorization that can not be translated to those in other ontologies. In this paper, we propose an alignment method between concept hierarchies by using a statical method. By using this method, a concept that exists in one hierarchy system but does not in the other can be located in a suitable position in the other. The key idea is to find similar categories between two system to be able to transfer concepts from one system to the other. Similarity is measured "κ statistic" based on instances belonging categories. The experiment of our method with concept hierarchies of Yahoo! and LYCOS results over 80% of accuracy to estimate appropriate positions of concepts between two hierarchies.

This paper is organized as follows: A concept hierarchy model discussed in the paper is defined in Section 2. Then we introduce our proposal method based on a learning method in order to align two concept hierarchies in Section 3. In the next section, the performance

of our system called HICAL (HIerarchical Concept ALignment system) based on our proposed method, is tested in a variety of settings, and we evaluated its results. We then discuss related work in regard to HICAL in Section 5, and present the conclusions of this study in Section 6.

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. Such information management system is comprised of 2 elements, i.e, categories and information instances. A category represents certain concepts and is used for classifying information instances. A category may have sub-categories, i.e., categories are connected to each other. An information instance represents a specific content of information and is expected to belong some categories. Figure 1 represents a concept hierarchy in which a black node denotes a category and a white node denotes an information instance. Note that our definition allows an instance belong to any categories including intermediate categories.

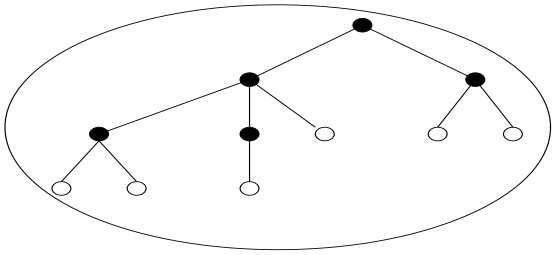

Figure 1: Concept Hierarchy Model

Now we are ready to consider our main problem. In Figure 2, 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 can be different in depth and size. 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 2, 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 learning rules for conversion from the concept hierarchy of C_1 to that of C_2, so that an instance that categorized in C_1 can subsequently be categorized in C_2. The 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 user can apply our method while continuing to use whichever concept hierarchy they are accustomed to.

3 Concept Alignment

At first we briefly describe our idea for concept alignment with an example. The example of our method is shown in Figure 3. Our method has 3 steps to transfer instances from a concept hierarchy C_1(source hierarchy) into the other concept hierarchy C_2(target hierarchy). First, our algorithm identifies similar categories using a statistical method. In the case of this example, the algorithm selects S_1 and S_2 as a similar category pair. Next, an alignment rule is constructed based on the similar category pair. In this case, the rule is constructed in the following form: "instances that are categorized in S_1 can be transfered into S_2". Each category in a source hierarchy is expected to have at most an aligning rule.

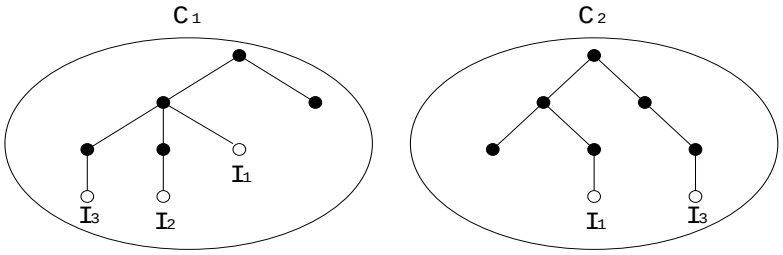

Figure 2: An Alignment Problem of Two Concept Hierarchies

Finally, the instances are transfered by following the alignment rules. In this example, three instances in S_1 are transfered by using this rule. In a case that the category does not have aligning rules, the rule of the upper categories may be used depending to evaluation policy (see Section 4.2).

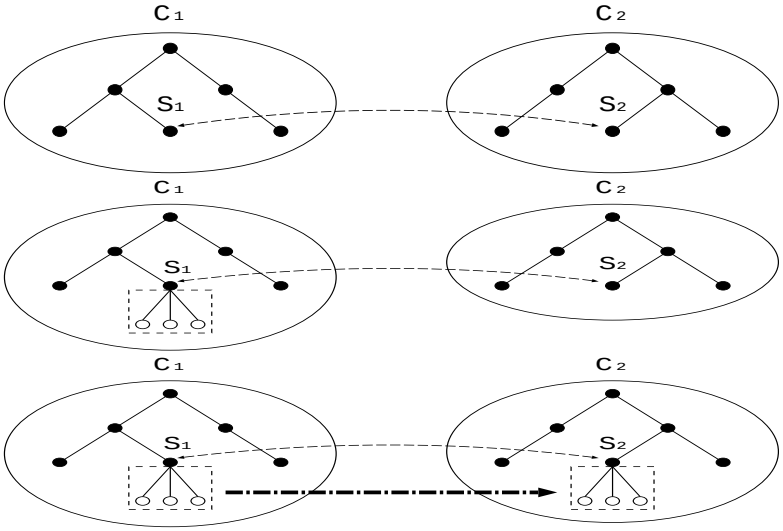

Figure 3: An Example of Transferring Instances using Concept Alignment

In our proposed method to generate alignment rules, we must first find categories that are similar to each other ("similar 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. 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. If the sets of instances for 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 99, then it is reasonable to generate an aligning rule for these categories, because they can be considered to have the same categorization criteria.

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 to evaluate 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

		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

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 4. 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 decide whether a particular instance in C_1 fits the concept hierarchy in C_2.

4 Experimental Evaluation

We developed a new alignment rule learning system called HICAL based on the proposed method. The following sections present experiments of the system.

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. These data were gathered in summer of 2000. The Yahoo! directory contains approximately 41,000 categories and 224,000 unique URLs. LYCOS contains approximately 5,700 categories and 48,000 unique URLs. Approximately 24,000 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 sub-categories) for experiments. 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

```
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 4: Alignment Algorithm

Table 2 illustrates the statistics on experimental data. It displays the number of categories and instances for each data.

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 5, 6 and 7. Data shown in these figures are average of 10-fold cross validation. "Exact rules" represent values of accuracy for a system that only uses the alignment rules for the category to which the instance belongs.

	Yahoo!		LYCOS		shared
	category	instance	category	instance	instance
Literature	493	3192	186	1119	468
Companies	7554	58609	413	5904	3992
Recreation	3164	19609	709	4941	1939

Table 2: Statistics on experimental data

"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 upper categories 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 whether concept hierarchies are rich or sparse in categorization. We tested two directions of alignment; from Yahoo! to LYCOS and from LYCOS to Yahoo!. The experimental result of former is shown in the left side of figures and the latter is shown in the right side.

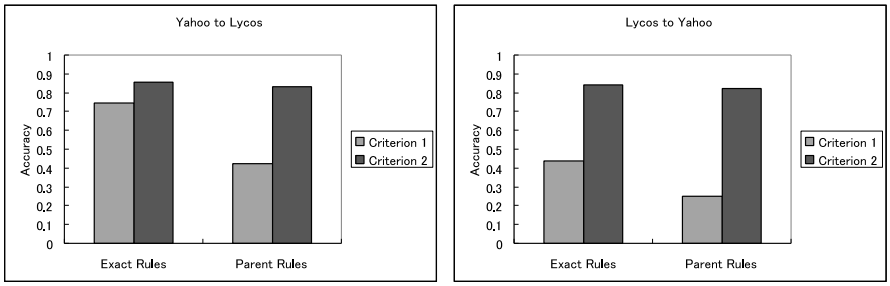

Figure 5: Result for Literature Domain

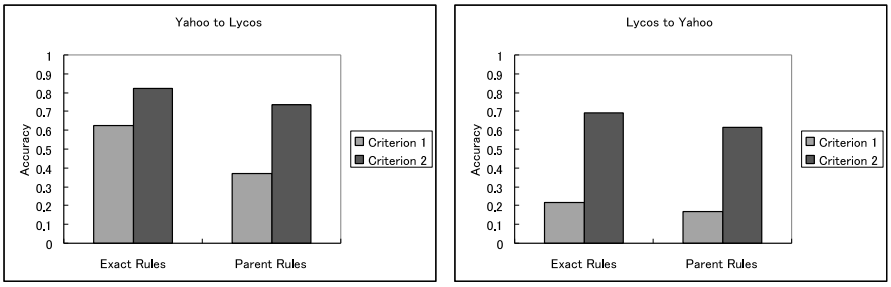

Figure 6: Result for Company Domain

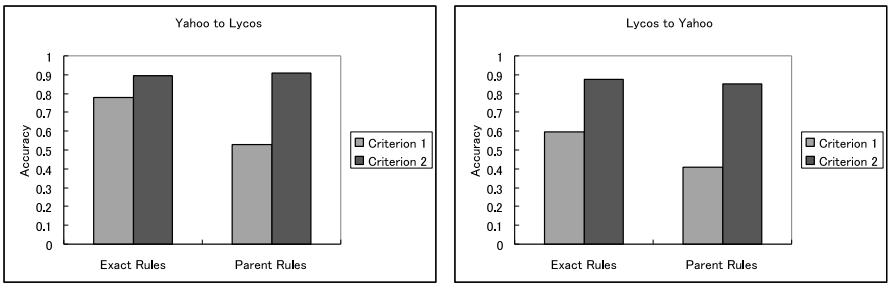

Figure 7: Result for Recreation Domain

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. Some observations are found. The first one is that Yahoo-to-Lycos alignment is more accurate than Lycos-to-Yahoo alignment. It is an expected result because depth and size of directory of Yahoo! is much more one in LYCOS. Generating rules to transfer instances in a richer directory to one in the other is generally easier than the inverse rules. For example, suppose concept hierarchy A contains category S within category X, however concept hierarchy B does not have any sub-categories under X. In this 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. The surprising fact is that difference of accuracy in both directions is nevertheless relatively small. It indicates that our method works properly even in this situation. 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 other finding is that using parent rules is not helpful to determine the appropriate position at least in our test cases. We expected that instances missing rules in their exactly belonging categories can use rules in more general categories instead. But this rule application was not correct even in "Criterion 2" except one test case. It is not what we expected. We should investigate rules generated with HICAL to know the reason.

The limitation of this method is that aligning concept hierarchies are expected to be similar in hierarchical structure. Aligning hierarchies with different categorizing policies would make unexpected results. For example, let's consider two concept hierarchies; one that classifies a food-related instance by type of foodstuff first, then by country of origin, and the other that classifies them by country of origin firstly and by type of food stuff secondly. In such a case, our current system HICAL would not work, because comparison between the two categories proceeds from general to specific (top to bottom). One solution for this problem is mixing bottom-up method with top-down method. It would need much larger search space and more computational cost but provide possibility of discovery of such complex relation between two hierarchies.

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 *et al.*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 *et al.*1997] and Blink [Blink 2000] are also similar to HICAL. The main difference is the use of hierarchies for categorization.

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

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 *et al.*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 two 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.

References

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

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

[Guarino *et al.*1995] Nicola Guarino and Pierdaniele Giaretta. Ontologies and Knowledge Bases, In *Towards Very Large Knowledge Bases*, IOS Press, Amsterdam, 1995.

[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 *et al.*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 *et al.*1999] Natalya F. Noy and Mark A. Musen. SMART: Automated Support for Ontology Merging and Alignment. In *Twelfth Banff Workshop on Knowledge Acquisition, Modeling, and Management*, Banff, Alberta, Canada. Available as SMI-1999-0813 ,1999.

[Noy *et al.*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, American Association for Artificial Intelligence, 2000.

[Rucker *et al.*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.