An Examination of the Relationships between Internet Directories

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

Abstract. Finding desired information on the internet is becoming increasingly difficult. Internet directories such as Yahoo! which organize web pages into hierarchical categories provides one solution to this problem, however, such directories are of limited use because some bias is applied both in the collection and categorization of the pages. Therefore, we propose a method for integrating multiple internet directories. Our method provides mapping of categories to be able to transfer documents from a directory to another, not simply merging two directories into one. Each document in a category of the source internet directory can be categorized into a similar category on the target internet directory. We present herein an effective algorithm for determining similar categories between two directories via a statistical method called κ-statistic. The benefits of the proposed method are twofold. First, unlike other methods, which use only a flat classification structure, the proposed method uses knowledge of hierarchies. Second, the proposed method does not use semantics of pages. The proposed method can determine the relationship between directories via hierarchical structure, whereas other methods often use semantics of pages such as keywords. In order to evaluate the proposed method, we conduct experiments using actual internet directories, Yahoo! and Google. The obtained results show that the proposed method achieves extensive improvements relative to both the Naive Bayes and Enhanced Naive Bayes approaches, without any text analysis on documents.

1 Introduction

The world-wide web (WWW) is now used not only by computer specialists, but by all sorts of people, from children to businessmen and businesswomen, which has resulted in an enormous quantity of web pages being available on the internet. This makes finding the pages containing the desired information rather difficult. Search engines are requisite to find desired information on the internet. Current search engines perform their functions via one of two methods in general: keyword search and directory search. A keyword search engine performs searches by means of user-specified keywords. Now keyword-based search engines like Google are reliable to find pages containing the specified keywords. However, if the user does not have a thorough knowledge of his/her search domain, the search engine is useless because the user cannot choose appropriate keywords. In such a situation, directory-based search engines do a good work. In a directory-based search engine, pages are evaluated and organized by humans before being registered within the search engines archive. Directory-based search engines provide knowledge for navigation of WWW. Users can reach the desired information in the issue of going up and down the directories.

Although pages are carefully selected and well-organized, a single internet directory is not sufficient because the search engine tends to have some bias in both collecting and categorizing pages. In order to solve these problems, we herein propose a method that coordinates multiple internet directories by estimating directory similarities. The proposed method does not simply merge multiple directories into a larger directory, but rather determines the relationship between directories. Many such public internet directories exist. Some are designed to cover wide domains, while others focus on special domains. Such internet directories are difficult to merge. Moreover, these directories should not be integrated, because existing the difference in concept hierarchies among directories is important when selecting and using such directory-based search engines. In this paper, we propose a method to solve this problem by determining rules by which to map categories in a directory to those in another directory. Our solution can be applicable not only to the internet directory problem, but also to integration of web marketplace catalogs [1][1].

This paper is organized as follows. In section 2, we define the problem of coordinating multiple internet directories and discuss related studies. Next, in section 3, we propose a new machine learning method for the above-mentioned problem. In section 4, we compare the performance of the proposed method to that of the Enhanced Naive Bayes approach [1] for an integration problem using real internet directories. Finally, in section 5, we present our conclusions.

2 Integration of multiple internet directories

In this section, we discuss the problem formalism and related research.

2.1 Problem specification

In order to state the problem, we introduce a model for the internet directories we intend to combine and assume two internet directories: a *source* directory and a *target* directory. The documents in the source internet directory are expected to be assigned to categories in the target internet directory. This produces a *virtually* integrated internet directory in which the documents in the source directory are expected to be members of both the source and target directories. This integrated directory inherits categorization hierarchy from the target internet directory.

[1] For example, integration of a distributor's catalog and a web marketplace.

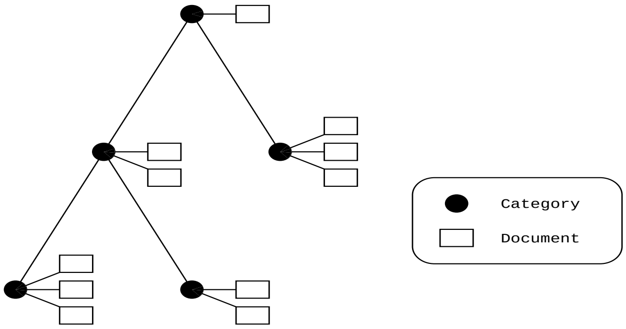

Fig. 1. Internet directory model

The internet directory model for source and target is as follows:

- The source internet directory, S_D contains a set of categories, $C_{s1}, C_{s2}, \ldots, C_{sn}$, that are organized into an "is-a" hierarchy. Each category contains documents.
- The target internet directory T_D contains a set of categories, $C_{t1}, C_{t2}, \ldots, C_{tm}$ that are organized into an "is-a" hierarchy. Each category contains documents.

The model can be represented as shown in Figure 1. The proposed model permits documents to be assigned to intermediate categories. This model is similar to the catalog model in [1], except that the categories are organized into a conceptual hierarchy. Since the catalog model ignores hierarchy structure and therefore cannot assign documents to an intermediate categories, our internet directory model is more general than the catalog model.

The problem addressed in this paper is to find an appropriate category C_t in the target directory T_D for each document D_{si} in the source directory S_D. An example is shown as mapping of a black box D_X in Figure 2. What we should do is to determine an appropriate category in T_D for a document which appears in S_D but *not* in T_D, because mapping is not necessary if the document is included in both the source and the target directories. Documents D_1 and D_2 in the Figure are examples of such documents. This mapping can have several possibilities, e.g., D_X can be mapped to an upper left category or to a lower left category, etc.

2.2 Related work

One popular approach to this kind of problem is to apply standard machine learning methods [7]. This requires a flattened class space having one class for every leaf node. The problem can then be considered as a normal classification problem for documents. Naive Bayes(NB) [9] is an established method used for

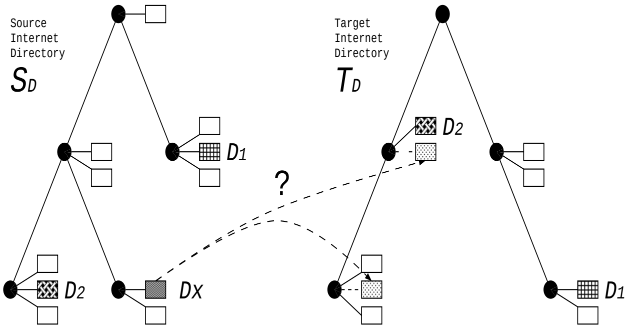

Fig. 2. Problem statement

this type of document classification framework. A classifier is constructed using words in documents. However, this classification scheme ignores the hierarchical structure of classes and, moreover, cannot use categorization information in the source directory. Enhanced Naive Bayes [1], hereafter referred to as Enhanced-NB, is a method which does use this information. Enhanced-NB will be discussed in the next section. Unlike NB, the systems in [6], [14] classify documents into hierarchical categories, and these systems use words in documents for classification rules. However, these systems cannot use categorization information in the source directory.

Another type of approach is ontology merging/alignment systems. These systems combine two ontologies which are represented in a hierarchal categorization. Chimaera [8] and PROMPT [11] are examples of such systems and assist in the combination of different ontologies. However, such systems require human interaction for merging or alignment. In addition to this requirement, these systems are based on similarity between words, which introduces instability. The dictionaries used for such system often have word similarity bias.

The bookmark-sharing systems of Siteseer [12] and Blink [2] also treat a similar problem. These systems attempt to share URL information, which appears in the source bookmark but not in the target bookmark. These systems flatten the categorization of bookmarks in the same manner as NB and determine the mapping of categories in each bookmark based on shared URL information. Such systems are problematic when a given URL does not fit into an exact category. kMedia [13] is another bookmark-sharing system that uses hierarchical structures explicitly but is dependent on similarity of words within pages. Bookmark-Agent [10] uses another approach, using bookmarks based on keywords. However, this system has the same problem as ontology merging/alignment systems.

2.3 Naive Bayes classification and enhancement thereof

Since we will be comparing Enhanced-NB approach and the proposed approach, we introduce NB and Enhanced-NB briefly in this section.

Naive Bayes is one method used to create classifiers from documents and their categories. The basic concept of the learned classifier is that the classifier assigns the category of maximum probability for the document we want to classify. When NB calculates the probability, the probability of keywords in the document are used as training data. In the context of the problem defined in section 2.1, the classifier is constructed using categories $C_{t1}, C_{t2}, \ldots, C_{tm}$ as well as keywords in documents $D_{t1}, D_{t2}, \ldots, D_{tv}$ in target internet directory T_D. The classifier is then applied to documents $D_{s1}, D_{s2}, \ldots, D_{su}$ in source internet directory S_D and assigns a category in T_D for each document.

As described above, NB does not use the categorization information $C_{s1}, C_{s2}, \ldots, C_{sn}$ in the source internet directory S_D. Enhanced-NB is an extension in which categorization information is used in order to increase accuracy. The basic concept behind this approach is that if documents belong to the same category in S_D, then the documents are more likely to belong to the same category in T_D. Enhanced-NB does not calculate conditional probability of category provided that only documents are given, but rather calculates conditional probability of category provided that both documents and categories in the source directory are given. Enhanced-NB then constructs a classifier to maximize the conditional probability. This treatment is more suitable to the problem stated in section 2.1. Agrawal and Srikant report in [1] an improvement over NB for a similar problem.

3 Determination of the relationship between categories of two internet directories

In this section, we explain our method in order to determine the relationship between categories in two internet directories. One characteristic of the proposed method is to use the hierarchical structure "as is". We use all categories including intermediate categories and leaf categories, because information for categorization can also be obtained from such intermediate categories. Another characteristic of the proposed method is reliance exclusively on the categorization structure of both directories, i.e., without semantic information of documents. We can determine the relationship between categories of two directories by statistically comparing the membership to categories of documents.

3.1 Basic concept

Although Enhanced-NB is developed for a very similar problem, it is missing an important feature, that is, categorization hierarchy. According to [1], the initial definition of the problem is identical to that of this paper. However, the previous paper assumes that "any documents assigned to an interior node really belong to a conceptual leaf node that is a child of that node" and concludes from this assumption that "we can flatten the hierarchy to a single level and treat it as a set of categories". However, the conclusion overlooks the categorization hierarchies. The categorizations are not independent of each other. The categorization hierarchies are usually structured using an "is-a" relation or other relations. If a

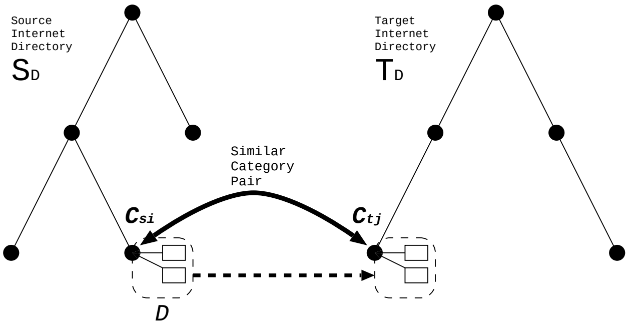

Fig. 3. Document transfer from source internet directory to target internet directory

document is categorized in a lower category in the categorization hierarchy, then the document should also be categorized in the upper categories [2]. If we flatten the categories, such information of relations between categories will be lost.

Another problem associated with NB is sensitivity to words in documents. For example, if a document contains the word *bank*, the category for the document could be a financial category; however, the category could also be a construction category. The quality of categorization with NB is thus not stable according to contents of documents due to natural language techniques how words are processed in it. It is critical to apply integration of internet directories because mixing of reliable internet directories maintained by human editors and such less reliable and stable machine-generated classifiers will make users confused and will hurt confidence for them. In addition to the problem of sensitivity, the analysis of the document to extract words is expensive.

We focus on similarity of the way of categorization not similarity of documents. Then how can we measure similarity of categorization? We utilize shard documents in the both internet directory as its measurement. If many documents in a category C_{si} also appears in another category C_{tj} at the same time, we regard that these two categories are similar, because the ways of categorization in C_{si} and C_{tj} are supposed to be similar, i.e., if another document D comes in C_{si}, it is likely that D will be also included in C_{tj}. This method allows both the keyword extraction process and the handling of word meaning to be avoided. The example shown in Figure 3 illustrates that documents in the bottom category in the source internet directory can be transferred into a similar category in the target internet directory.

3.2 κ-statistic

The remaining problem is how to determine pairs of similar categories. In order to establish similar category pairs, we adopt a statistical method to deter-

[2] Note that the reverse is not always true because documents can be categorized to intermediate categories.

Table 1. Classification of documents by two categories

		Category C_{tj}	
		belong	not belong
Category C_{si}	belong	N_{11}	N_{12}
	not belong	N_{21}	N_{22}

mine the degree of similarity between two categorization criteria. The κ-statistic method [4] is an established method to evaluate similarity between two criteria. Suppose there are two categorization criteria, C_{si} in S_D and C_{tj} in T_D. We can determine whether a particular document belongs to a particular category or not[3]. Consequently, documents are divided into four classes, as shown in Table 1. The symbols N_{11}, N_{12}, N_{21} and N_{22} denote the numbers of documents for these classes. For example, N_{11} denotes the number of documents which belong to both C_{si} and C_{tj}. We may logically assume that if categories C_{si} and C_{tj} have the same criterion of categorization, then N_{12} and N_{21} are nearly zero, and if the two categories have a different criterion of categorization, then N_{11} and N_{22} are nearly zero. The κ-statistic method uses this principle to determine the similarity of categorization criteria.

In the κ-statistic, we calculate the probability P, which denotes the percentage of coincidence of the conceptual criteria, and the probability P' which denotes the percentage of coincidence of the conceptual criteria by chance.

$$P = \frac{N_{11} + N_{22}}{N_{11} + N_{12} + N_{21} + N_{22}}$$

$$P' = \frac{(N_{11} + N_{12})(N_{11} + N_{21}) + (N_{21} + N_{22})(N_{12} + N_{22})}{(N_{11} + N_{12} + N_{21} + N_{22})^2}$$

As such, the value of the κ-statistic is represented as the following equation:

$$\kappa = \frac{P - P'}{1 - P'}$$

Next, we examine whether we can assume the $\kappa = 0$, which indicates that the percentage of coincidence of two conceptual criteria is zero. We therefore calculate test statistic Z according to the following equation.

$$Z = \kappa \sqrt{\frac{(N_{11} + N_{12} + N_{21} + N_{22})(1 - P')}{P'}} \tag{1}$$

The value Z follows a normal distribution. A null hypothesis is considered to occur when the percentage of coincidence of concept criteria is zero. When we

[3] Remember that categorization in a directory is determined using the nodal structure of categorization hierarchy.

assume a significance level of 5% and the following equation is satisfied, we can dismiss the null hypothesis.

$$Z \geq 1.64486$$

When the null hypothesis can be dismissed, the criteria are determined to be the same.

3.3 Determination of pairs of similar categories

The relationship between the two categorization criteria is examined from top to bottom. This algorithm is shown in Figure 4. First, the most general categories in the two categorization hierarchies are compared using the κ-statistic. If the comparison confirms that the two categories are similar, then the algorithm outputs this pair of categories. At the same time, the algorithm generates all possible pairs of their children categories. It generates two lists of categories each of which is a list of the confirmed category and its children categories, then picks up one from each and makes a pair except the original pair. This new pair is then evaluated recursively using the κ-statistic method. When a similar pair is not generated, the algorithm outputs the rules between the two categorization hierarchies.

This top-down comparing approach attempts to reduce the exploration space using the structure of both directories as a guide. Assuming that similar categories include similar sub-categories, we can skip the combination of categories which is likely unnecessary[4]. This is another benefit of using hierarchy “as is”, in contrast to the flattening approach.

3.4 Application policy of rules

Since the proposed method uses categorization similarity, if a category in the source internet directory does not have a similar category in the target directory, then those documents cannot be categorized in the target internet directory. In order to avoid this problem, we once again use categorization hierarchy. If the system can not find a similar category pair for the category containing the document, it applies the rule generated for the parent category instead.

Let us explain this process by using an example. Figure 5 illustrates the above-mentioned situation. The bottom category in the source internet directory has no similar category and the parent category has a similar category represented by the broken line. In this case, the system applies the rule for the parent category to the document in the bottom category. As a result, the documents are assigned to a category in target internet directory according to the rule for its parent category.

[4] This assumption is sometimes too strong, e.g., a child category of a category is not so relevant, whereas its child category (a grandchild category of the original category) is relevant. In such cases, we should relax this constraint when exploring future candidates.

```
main_algorithm
Input:    $C_{s1}$,   // Top category in $S_D$
          $C_{t1}$,   // Top category in $T_D$
Output:   $R$;        // Set of similar category pair
begin
    $t := 1$;
    $R := \phi$;
    $X_1 := [[C_{s1}, C_{t1}]]$;
    while $X_t \neq \phi$
        while $X_t \neq \phi$
            $I := element\ in\ X_t$;
            $C_1, C_2 := two\ categories\ in\ I$;
            if $C_1$ and $C_2$ is similar
                $X_{t+1} := X_{t+1} + make\_comb(C_1, C_2)$;
                $R := R + I$;
            fi;
            $X_t := X_t - I$;
        end;
        $t := t + 1$;
    end;
    return $R$;
end;
```

```
make_comb
Input:    $C_s$,   // Category in $S_D$
          $C_t$,   // Category in $T_D$
Output:   $S$;     // Set of category pair in $S_D$ and $T_D$
begin
  $S$ := set of category pair, $C_s$ and childs in $C_t$;
  $S$ := $S$ + set of category pair, $C_t$ and childs in $C_s$;
  $S$ := $S$ + set of category pair made by combination
             of childs in $C_s$ and childs in $C_t$;
  return $S$;
end;
```

Fig. 4. Algorithm to determine similar category pair

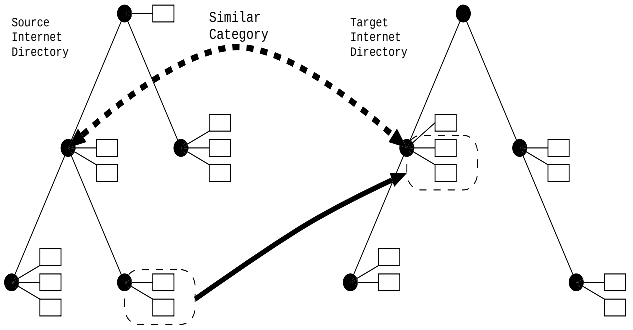

Fig. 5. An example of rules for a category which has no similar category in the target internet directory

3.5 Computational complexity

In this section, we discuss the computational complexity of the proposed algorithm shown in Figure 4. Suppose that two concept hierarchies S_D and T_D are trees of width b (the number of subcategories for each categories) and height d (the number of categories from the top to the bottom) for sake of simplicity. In this case, we calculate the total number of categories N using the following equation:

$$N = b^0 + b^1 + \ldots + b^d$$

Comparison of similarity among all categories requires $N \times N$ comparisons. As a result, the order of computational complexity is $O(b^{2d})$.

Next, we consider the cost of our algorithm. Assuming that after the similarity tests using κ-statistic have been conducted, $n\%$ of the tests are confirmed to be similar to other categories. The number of pairs produced by *make_comb* is $(b+1)^2 - 1$ because it makes pairs from two lists each of which consists of the category itself and its child categories except the orignal pair. The number of expected combination pairs after processing of the κ-statistic is $n \times (b^2 + 2b) + (1-n) \times 0$. The formula $(1-n) \times 0$ arises because no similar pair is produced when the κ-statistic does not confirm the similarity. We finally obtain the total number of nodes M to be compared as the product of the number of examined nodes and the expected number of nodes. M is given by the following equation:

$$\begin{aligned} M = {} & b^0 + n^1(b^2 + 2b)^1 \\ & + n^2(b^2 + 2b)^2 + \ldots + n^d(b^2 + 2b)^d \end{aligned}$$

The order of complexity is then obtained as $O(n^d \times b^{2d}) = O((n \times b^2)^d)$. Since n denotes the percentage of category pairs confirmed to be similar, $0 \leq$

Table 2. Statistics on the experimental data

	Yahoo!		Google		shared
	categories	links	categories	links	links
Autos	782	5200	583	7909	1002
Movies	4001	16947	4623	21244	2735
Outdoors	2247	13932	825	13725	716
Photography	375	4173	170	3999	536
Software	997	4686	2142	35628	952

$n \leq 1$ is satisfied. As a result, we obtain less computational complexity using the algorithm in the Figure 4. In other words, although the depth remains the same, the search algorithm can reduce computation by constraining the breath size.

4 Experiments using internet directories

4.1 Experimental settings

In order to evaluate proposed algorithm, we conducted experiments using data collected from the Yahoo! [15] and Google [5][5] internet directories. The data was collected in the fall of 2001. In order to compare the proposed method to that in [1], we selected the same locations in Yahoo! and Google for the experiment data. The locations are as follows:

- Yahoo! : Recreation / Automotive
  Google : Recreation / Autos
- Yahoo! : Entertainment / Movies_and_Film
  Google : Arts / Movies
- Yahoo! : Recreation / Outdoors
  Google : Recreation / Outdoors
- Yahoo! : Arts / Visual_Arts / Photography
  Google : Arts /Photography
- Yahoo! : Computers_and_Internet / Software
  Google : Computers / Software

Table 2 shows the numbers of categories, links in each internet directory and links included in both internet directories. The links are considered to be the same when the URLs in both directories are identical.

We conducted 10-fold cross validation for shared links. Shared links were divided into 10 data sets; nine of these sets were used to construct rules, and the remaining set was used for testing. Ten experiments were conducted for each

[5] Since data in Google is constructed by the data in dmoz [3], we collected data through dmoz.

Table 3. Results of Yahoo! as the source internet directory and Google as the target internet directory

Dataset	Accuracy		Improvement
	Enhanced-NB [1]	SBI	
Autos	76.2	88.0	11.8
Movies	42.6	77.9	35.3
Outdoors	77.8	74.0	-3.8
Photography	72.8	79.7	6.9
Software	62.4	73.4	11.0
Average	64.4	78.4	14.0

data set, and the average accuracy is shown in the result. In order to compare the proposed method to the Enhanced-NB approach, the classifiers are assumed to correctly assign documents when the document is categorized either in the same category as the test data or in the parent categories of the test data. Enhanced-NB constructs classifiers for only the first-level categories in the experimental domain, whereas the proposed method uses all of the categories from top to bottom. The significance level for the κ-statistic was set at 5%.

4.2 Experimental Results

The experimental results are shown in Tables 3 and 4. Table 3 shows the results obtained using Yahoo! as the source internet directory and Google as the target internet directory, and Table 4 shows the results obtained using Google as the source internet directory and Yahoo! as the target internet directory. For comparison, these tables also include the results of [1]. Enhanced-NB indicates the method of [1] and SBI indicates the similarity-based integration method, proposed in this paper. The data obtained for the proposed method and that presented in [1] are not truly comparable because the collection date is different[6].

The proposed algorithm did well enough compared to Enhanced-NB, performing more than 10% better in accuracy than Enhanced-NB. In the Movies domain, the proposed algorithm performs much better in accuracy than Enhanced-NB. One reason for this is that pages related to movies contain various words to indicate numerous movie settings, making classification using word based systems difficult. On the other hand, in the Outdoors domain, performance is similar for both methods because the Outdoors domain treats pages using special outdoors-related words, and no classification method for the Outdoors domain has been established for human readers.

As mentioned earlier, in the experiments described in [1], the classifiers induced by the system have the ability to classify the documents into only the first-

[6] In addition, differences may have occurred due to data selection. For example, Yahoo! has links that use @(the at mark). The treatment of such links was not discussed in [1]. In the present study, we did not use such links.

Table 4. Results of Google as the source internet directory and Yahoo! as the target internet directory

Dataset	Accuracy		Improvement
	Enhanced-NB [1]	SBI	
Autos	73.1	83.6	10.5
Movies	46.2	68.1	21.9
Outdoors	65.4	68.9	3.5
Photography	51.3	60.4	9.1
Software	58.6	74.0	15.4
Average	58.9	71.0	12.1

level categories of the test domain in the target internet directory. The proposed classifiers can classify the documents into sub-categories as well. For example, we used 4,623 categories for document assignment in the Movies domain of Yahoo! as the source internet directory and Google as the target internet directory, whereas their system used only 40 categories. Therefore, the classifiers obtained by the proposed method are more reliable and useful than those obtained by other methods.

5 Conclusions

In this paper, a statistical based technique was proposed for integrating multiple internet directories by determining the relationship between these directories. The proposed method uses κ-statistics to find similar category pairs, and transfers the document categorization from a category in the source internet directory into a similar category in the target internet directory. The proposed method has an advantage in document treatment, whereby it relies on only the category structure, not words or word similarity in a document. The performance of the proposed method was tested using actual internet directories, and the results of these tests show that the performance of the proposed method was more than 10% in accuracy better than that of the previous method even though the number of categories are by far larger.

Although the present results are encouraging, much has yet to be done. The limitation of the proposed method is that this method is based on the existence of shared links. In other words, the proposed method is less reliable if there are fewer shared links. To improve it, various methods might be used to obtain semantic information in order to increase the number of shared links. e.g., regarding similar pages as the same links. Moreover, the present method can find only one-to-one mapping rules. If the system is able to find a set of categories in the source directory and a set of categories in the target directory that are similar, the categorization hierarchies can be divided into smaller parts. Finally, the proposed method should be expanded so as to apply to more than three concept hierarchies. In such a case, despite the conflict between several concept hierarchies, more information is expected to be obtained from other concept hierarchies. The aforementioned tasks for improvement will be investigated in future studies.

References

1. R. Agrawal and R. Srikant. On integrating catalogs. In *Proceedings of the Tenth International World Wide Web Conference (WWW-10)*, pages 603–612, 2001.
2. Blink. http://www.blink.com/, 2000.
3. dmoz. http://dmoz.org/, 2001.
4. J. L. Fleiss. *Statistical Methods for Rates and Proportions.* John Wiley & Sons, 1973.
5. Google. http://directory.google.com/, 2001.
6. D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In D. H. Fisher, editor, *Proceedings of the 14th International Conference on Machine Learning*, pages 170–178, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.
7. P. Langley. *Elements of Machine Learning.* Morgan Kaufmann, 1996.
8. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of the Conference on Principiles of Knowledge Representation and Reasoning (KR-00)*, pages 483–493, S.F., Apr. 11–15 2000. Morgan Kaufman Publishers.
9. T. M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.
10. M. Mori and S. Yamada. Bookmark-agent: Information sharing of urls. In *Poster Proceedings of the 8th International World Wide Web Conference, WWW-8*, pages 70–71, 1999.
11. N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000)*, pages 450–455, Menlo Park, 2000. AAAI Press.
12. J. Rucker and M. J. Polanco. Siteseer: Personalized navigation for the web. *Communications of the ACM*, 40(3):73–75, 1997.
13. H. Takeda, T. Matsuzuka, and Y. Taniguchi. Discovery of shared topics networks among people – a simple approach to find community knowledge from www bookmarks. In R. Mizoguchi and J. Slaney, editors, *Proceedings of the 7th Pacific Rim International Conference on Topics in Artificial Intelligence (PRICAI-2000)*, volume 1886 of *LNAI*, pages 668–678, Berlin, 2000. Springer.
14. K. Wang, S. Zhou, and S. C. Liew. Building hierarchical classifiers using class proximity. In M. Atkinson, M. E. Orlowska, P. Valduriez, S. Zdonik, and M. Brodie, editors, *Proceedings of the 25th international Conference on Very Large Data Bases*, pages 363–374, Los Altos, CA 94022, USA, 1999. Morgan Kaufmann Publishers.
15. Yahoo! http://www.yahoo.com/, 2001.