Discovering Relationships Among Catalogs

Ryutaro Ichise[1,2], Masahiro Hamasaki[2], and Hideaki Takeda[1,2]

[1] National Institute of Informatics, Tokyo 101-8430, Japan
[2] The Graduate University for Advanced Studies, Tokyo 101-8430, Japan
{ichise,takeda}@nii.ac.jp, hamasaki@grad.nii.ac.jp

Abstract. When we have a large amount of information, we usually use categories with a hierarchy, in which all information is assigned. The Yahoo! Internet directory is one such example. This paper proposes a new method of integrating two catalogs with hierarchical categories. The proposed method uses not only the contents of information but also the structures of both hierarchical categories. In order to evaluate the proposed method, we conducted experiments using two actual Internet directories, Yahoo! and Google. The results show improved performance compared with the previous approaches.

1 Introduction

The progress of information technologies has enabled us to access a large amount of information. This naturally demands a method of managing such information. One popular method of information management systems is to utilize a concept hierarchy and categorize all information into the concept hierarchy. Examples include catalogs of publications, file directories, Internet directories and shopping catalogs. A catalog of publications uses one standardized concept hierarchy for categorization. However, most concept hierarchies for information management are hard to standardize because each concept hierarchy has its own purpose and user. This situation produces difficulties when we use multiple information sources with different concept hierarchies. Hence, technologies for integrating multiple catalogs with different concept hierarchies are necessary for seamless access among them. Similarity-based integration (SBI) [7] is an effective method for solving this problem. The main idea of SBI is to utilize only the similarity of categorizations across concept hierarchies. Namely, SBI does not analyze the contents of information assigned to the concept hierarchies. In this paper, we propose an extension of SBI which uses the contents in information instances. The basic idea of our approach is to combine SBI and Naive Bayes [12].

This paper is organized as follows. Section 2 characterizes the catalog integration problem, which is the subject of this paper. Section 3 describes related work. Section 4 presents new catalog integration algorithms and mechanisms. Section 5 applies the algorithms to real world catalogs, which are Internet directories, to demonstrate our algorithm's performance, and discusses our experimental results and methods. Finally, in Section 6 we present our conclusions and future work.

E. Suzuki and S. Arikawa (Eds.): DS 2004, LNAI 3245, pp. 371–379, 2004.

2 Catalog Integration Problem

In order to state our problem, we introduce a model for the catalogs we intend to integrate. We assume there are two catalogs: a *source* catalog and a *target* catalog. The information instances in the source catalog are expected to be assigned to categories in the target catalog. This produces a *virtually* integrated catalog in which the information instances in the source catalog are expected to be members of both the source and target catalogs. This integrated catalog inherits the categorization hierarchy from the target catalog.

Our catalog model for the source and target is as follows:

- The source catalog S_C contains a set of categories $C_{s1}, C_{s2}, \ldots, C_{sn}$ that are organized into an "is-a" hierarchy. Each category can also contain information instances.
- The target catalog T_C contains a set of categories $C_{t1}, C_{t2}, \ldots, C_{tm}$ that are organized into an "is-a" hierarchy. Each category can also contain information instances.

We will assume that all information instances in both catalogs are supposed to have some attributes for each. The problem addressed in this paper is finding an appropriate category C_t in the target catalog T_C for each information instance I_{si} in the source catalog S_C. What we need to do is determine an appropriate category in T_C for an information instance which appears in S_C but *not* in T_C, because mapping is not necessary if the information instance is included in both the source and the target catalogs.

3 Related Work

One popular approach to this kind of problem is to apply standard machine learning methods. This requires a flattened class space which has one class for every leaf node. Naive Bayes (NB) [12] is an established method used for this type of instance classification framework. However, this classification scheme ignores the hierarchical structure of classes and, moreover, cannot use the categorization information in the source catalog. Enhanced Naive Bayes (E-NB) [1] is a method which does use this information. Although E-NB has a better performance than NB, it does not achieve the performance of SBI [7]. SBI will be discussed in the next section. GLUE [4] is another type of system employing NB. GLUE combines NB and a constraint optimization technique. Unlike normal NB, the document classification systems in [8, 10, 16] classify documents into hierarchical categories, and these systems use words in the documents for classification rules. These systems can be applicable to the catalog integration problem. However, these systems cannot use the categorization information in the source catalog.

Another type of approach is ontology merging/alignment systems. These systems combine two ontologies, which are represented in a hierarchal categorization. Chimaera [11] and PROMPT [13] are examples of such systems and assist in the combination of different ontologies. However, such systems require human

interaction for merging or alignment. FCA-Merge [15] is another type of ontology merging method. It uses the attributes of concepts to merge different ontologies. As a result, it creates a new concept without regarding the original concepts in both ontologies.

From the viewpoint of catalog integration, an approach besides E-NB is to construct the abstract-level structure of two hierarchies [14]. This approach does not direct the transformation of source and target information, but transforms the information via the abstract-level structure. It is relatively easy to transfer information through many hierarchical structures, but it is hard to create a common structure for these hierarchies.

4 A Multi-strategy Approach

In order to solve the problem stated in Section 2, we proposed the SBI system [7]. SBI used only the similarity of the categorization of instances. In other words, SBI did not use the contents of the instances. In this paper, we propose a method which combines the SBI approach and the NB approach. This approach naturally implies that the new approach is able to use more information than the previous approaches; thus, we can expect to obtain a better performance compared with previous methods. In this section, we briefly explain the SBI and NB methods, and then we show the new combination approach, called SBI-NB.

4.1 Similarity-Based Integration

SBI focuses on the similarity of the way of categorization, not on the similarity of instances. Then, how do we measure the similarity of categorization? We utilize the shared instances in both the source and target catalogs as our measurement standard. If many instances in category C_{si} also appear in category C_{tj} at the same time, we consider these two categories to be similar, because the ways of categorization in C_{si} and C_{tj} are supposed to be similar, i.e., if another instance I is in C_{si}, it is likely that I will also be included in C_{tj}.

SBI adopts a statistical method to determine the degree of similarity between two categorization criteria. The κ-statistic method [5] is an established method for evaluating the similarity between two criteria. Suppose there are two categorization criteria, C_{si} in S_C and C_{tj} in T_C. We can determine whether or not a particular instance belongs to a particular category. Consequently, instances are divided into four classes, as shown in Table 1. The symbols N_{11}, N_{12}, N_{21} and N_{22} denote the number of instances for these classes. For example, N_{11} denotes the number of instances 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 the categorization criteria. If you are interested in more details of the SBI mechanism, please refer to the work of [7].

Table 1. Classification of instances by two categories.

		Category C_{tj}	
		Belongs	Not belongs
Category C_{si}	Belongs	N_{11}	N_{12}
	Not belongs	N_{21}	N_{22}

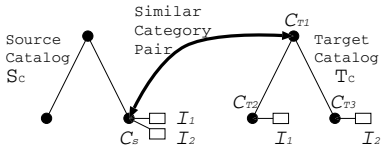


Fig. 1. An example of a learned rule.

4.2 SBI-NB

A problem of SBI is that it is hard to learn a mapping rule when the destination category is in a lower category in the target concept hierarchy. In other words, the learned rules are likely to assign relatively general categories in the target catalog. Let us consider an example of this problem. Assume that a category in the source directory has two instances, and these instances are categorized in different categories in the target catalogs. We also assume that the two categories in the target catalog have the same parent category. This situation is shown in Figure 1. In this case, the SBI system could induce a mapping rule between C_s and C_{T1}. According to the problem definition, this rule is reasonable since we assume that the hierarchy relationship is "is-a" only. However, the best rules for these instances are rules which categorize them directly into category C_{T2} and C_{T3}. In order to induce this type of rules, we propose to combine a contents-based classification method after we apply the SBI algorithm. Since NB is very popular and easy to use, we adopt NB as the contents-based classification method.

The NB method is used to create classifiers from instances and their categories. The basic concept of the learned classifier is that the classifier assigns the category of maximum probability for the instance we want to classify. In the context of the problem defined in Section 2, the classifier is constructed using categories $C_{t1}, C_{t2}, \ldots, C_{tm}$ as well as the attribute values in instances $I_{t1}, I_{t2}, \ldots, I_{tv}$ in target catalog T_C. The classifier is then applied to instances $I_{s1}, I_{s2}, \ldots, I_{su}$ in source catalog S_C and assigns a category in T_C for each instance. On the other hand, in order to apply the NB algorithm for hierarchical classification, we utilize the simple method of the *Pachinko Machine* NB [10]. The Pachinko Machine classifies instances at internal nodes of the tree, and greedily selects sub-branches until it reaches a leaf [10]. In the Pachinko Machine NB method, when the system selects categories in the internal node, NB is used as the classification method. In addition, our system makes a virtual category in an internal node, and treats it in the same manner as a normal category. The virtual category has instances assigned to the internal node. This is because of the capability of classification for the internal node. This method is applied after the rule induced by SBI decides the starting category for the Pachinko Machine NB. To implement our system, we utilize the NB system called Rainbow [9] and SBI.

Table 2. Statistics on the experimental data.

	Yahoo!		Google		Shared
	Categories	Instances	Categories	Instances	instances
Autos	885	5134	874	9702	544
Movies	5297	19192	7947	36288	1480
Outdoors	2590	7960	1221	17065	362
Photography	578	5548	278	5443	305
Software	513	3268	2339	41883	353

5 Experiment

5.1 Experimental Settings

In order to evaluate the proposed algorithm, we conducted experiments using real Internet directories collected from Yahoo! [17] and Google [6][1]. The locations in Yahoo! and Google 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, the instances in each Internet directory and the instances included in both Internet directories. From Table 2, we can see that each Internet directory tends to have its own bias in both collecting and categorizing pages. This fact proves the necessity for catalog integration.

We conducted ten-fold cross validations for the shared instances. Ten experiments were conducted for each data set, and the average accuracy is shown in the results. We compared the SBI-NB system with the SBI system and the NB system with flattened classes. Keyword extraction from documents and keyword indexing are necessary to use NB. In our experiment, we obtained web pages from the Internet and indexed all of them by using Rainbow. The same indexing method was used for NB and SBI-NB. The accuracy is measured for each depth of the Internet directories. If the correct categories are deeper than the depth which is used in the experiment, the bottommost categories are considered as the answers instead of the actual answer. In our experimental settings, other categories, such as a parent of the actual answer, can also be a correct category in semantic meaning. However, in this experiment, we utilized strict answer criteria[2]. The number of categories in the Internet directories is shown in Table 3. As one can see from this table, categorization for a lower category is more difficult than it is for upper categories. The significance level for the κ-statistic was set at 5%.

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

[2] The criteria of accuracy in this paper is more strict than that in the experiment of [7]. In the previous experiment, we chose other criteria to compare the result with those of another system. Therefore, the accuracy in this paper looks lower than that in [7] because of different criteria.

Table 3. Number of classes at each depth.

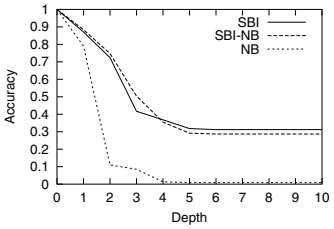
Depth	Autos		Movies		Outdoors		Photography		Software	
	Yahoo!	Google	Yahoo!	Google	Yahoo!	Google	Yahoo!	Google	Yahoo!	Google
1	37	15	38	31	46	27	31	10	32	89
2	240	227	214	202	224	173	83	39	125	494
3	475	690	730	5345	541	441	328	162	258	1154
4	765	833	3659	7239	1286	748	578	267	403	1817
5	867	852	4979	7789	2120	1112	578	277	489	2171
6	883	874	5277	7947	2391	1221	578	278	507	2316
7	885	874	5293	7947	2590	1221	578	278	513	2337
8	885	874	5296	7947	2590	1221	578	278	513	2339
9	885	874	5297	7947	2590	1221	578	278	513	2339
10	885	874	5297	7947	2590	1221	578	278	513	2339

5.2 Experimental Results

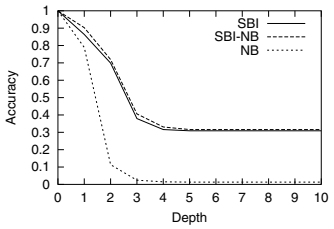
The experimental results are shown in Figure 2. The vertical axes show the accuracy and horizontal axes show the depth of the concept hierarchies. The experimental domains of the graphs are Autos, Movies, Outdoors, Photography and Software, from top to bottom. The left side of Figure 2 shows the results obtained using Google as the source catalog and Yahoo! as the target catalog, and the right side of Figure 2 shows the results obtained using Yahoo! as the source catalog and Google as the target catalog. For comparison, these graphs also include the results of SBI and NB. SBI-NB denotes the results of the method proposed in this paper. Since it is impossible to calculate the accuracy in the Movie domain on Yahoo! as a target catalog by using NB because of the large number of categories, the accuracy is not shown in Figure 2.

The proposed SBI-NB algorithm has high accuracy compared with NB for all experimental domains. In particular, our algorithm finds a solution in the Movie domain for Yahoo! whereas NB cannot find a solution. From this we can conclude that our algorithm is better than the NB method. On the other hand, the proposed algorithm is better than SBI for all domains except Automotive and Outdoors for Yahoo!. In addition, in these two domains, the difference of accuracy between our algorithm and SBI is very small. Hence, we can conclude that our algorithm is better than SBI. From the comparison of the results of both NB and SBI with SBI-NB, we can expect high accuracy when NB produces a good result. For example, in the Photography domain, the proposed algorithm performs much better in accuracy than the original SBI. One reason for this is that the NB works well. In other words, the contents-based classification is suited for this domain. Our proposed algorithm effectively combines the contents-based method with the category similarity-based method. On the other hand, when the NB method has poor accuracy, our method has at least the same performance of SBI and does not have any side effect from NB. For example, in the Autos domain, the performance is similar for SBI and SBI-NB, regardless of the poor performance of NB. Since the SBI-NB method proposed in this paper has a high performance in many domains compared with NB and SBI, we can conclude that our approach is a good method for integrating the approach of document classification based on the attribute of an instance and the approach of the similarity-based integration method.

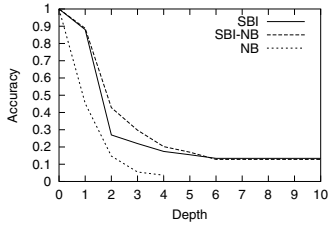
Integration into Yahoo!: Automotive



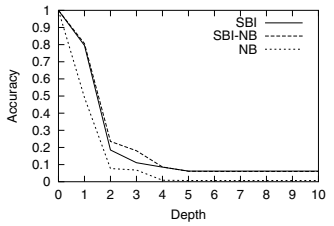
Integration into Google: Autos



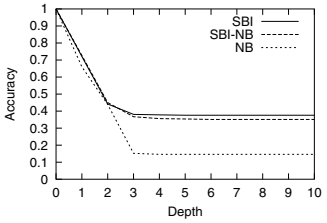
Integration into Yahoo!: Movies_and_Film



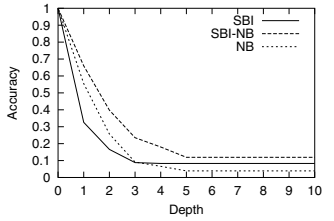
Integration into Google: Movies



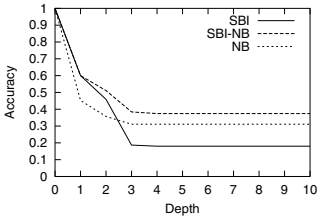
Integration into Yahoo!: Outdoors



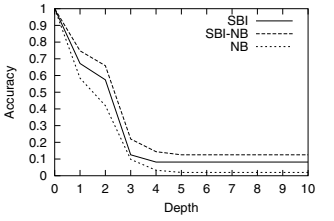
Integration into Google: Outdoors



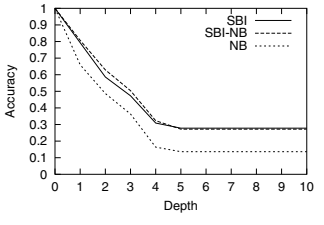
Integration into Yahoo!: Photography



Integration into Google: Photography



Integration into Yahoo!: Software



Integration into Google: Software

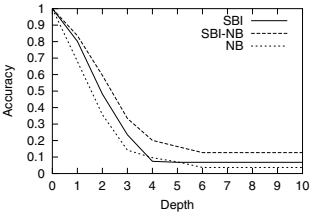


Fig. 2. Experimental Results.

6 Conclusion

In this paper, a new technique was proposed for integrating multiple catalogs. The proposed method uses not only the similarity of the categorization of catalogs but also the contents of information instances. 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 is more accurate for most of the experiments.

Although the present results are encouraging, much has yet to be done. For this research, we applied the Pachinko NB method as the contents-based classification method. However, other methods such as SVM [2] and shrinkage [10] are adoptable for our system because of the independency gained from the SBI method. We plan to test such combinations. Other future work includes expanding the proposed method so that it can apply to more than three catalogs.

Acknowledgment

We would like to thank Dr. McCallum for providing the Naive Bayes program.

References

1. R. Agrawal and R. Srikant. On integrating catalogs. In *Proc. of the Tenth Int. WWW Conf.*, pp. 603–612, 2001.
2. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.
3. dmoz. http://dmoz.org/, 2003.
4. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proc. of the 11th Int. WWW Conf.*, 2002.
5. J. Fleiss. *Statistical Methods for Rates and Proportions.* John Wiley & Sons, 1973.
6. Google. http://directory.google.com/, 2003.
7. R. Ichise, H. Takeda and S. Honiden. Integrating multiple internet directories by instance-based learning. In *Proc. of the 18th Int. Joint Conf. on AI*, pp. 22-28, 2003.
8. D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proc. of the 14th Int. Conf. on Machine Learning*, pp. 170–178, 1997.
9. A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, http://www.cs.cmu.edu/ mccallum/bow/, 1996
10. A. K. McCallum, R. Rosenfeld, T. M. Mitchell and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proc. of the 15th Int. Conf. on Machine Learning*, pp. 359–367, 1998.
11. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proc. of the Conf. on Principles of Knowledge Representation and Reasoning*, pp. 483–493, 2000.
12. T. M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.
13. N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proc. of the 17th National Conf. on AI*, pp. 450-455, 2000.

14. B. Omelayenko and D. Fensel. An analysis of B2B catalogue integration problems. In *Proc. of the Int. Conf. on Enterprise Information Systems*, pp. 945–952, 2001.
15. G. Stumme and A. Madche. FCA-Merge: Bottom-up merging of ontologies. In *Proc. of the 17th Int. Joint Conf. on AI*, pp. 225–230, 2001.
16. A. Sun and E. Lim. Hierarchical Text Classification and Evaluation. In *Proc. of IEEE Int. Conf. on Data Mining*, pp. 521–528, 2001.
17. Yahoo! http://www.yahoo.com/, 2003.