Category Mapping for Integration of Web Shopping Search

Jyh-Jong Tsay Tzung-Bang Chen Chih-Hsiang Lin

Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi, Taiwan 62107, ROC.
{tsay, ctp94}@cs.ccu.edu.tw

December 10, 2007

Abstract

As there are more and more online stores and shopping sites available on the web, it becomes very time-consuming, if not impossible, to visit and search all the online stores and shopping sites to find the best product and store to buy. There is great need to build an integrated shopping search system that provides a unified interface and metasearch capability to search and access all the shopping sites in one query submission. Observe that most of the online store and shopping sites provide a category directory and a query interface for users to find products of their interest. One usually navigates along the category directory to select a category, and then submits a query to find products in the selected category whose descriptions match the query. Automatic selection of categories from each shopping site is one of the fundamental problems to build an integrated shopping search systems. Note that category selection reduces the space to search, and improves the query time and quality of the query result. Quick and high quality response from each shopping site is very crucial to the performance of an integrated shopping search system. In this paper, we study the problem of automatic category selection. In particular, we study the problem of mapping categories among shopping web sites. We formulate the mapping problem as a search problem, and present an efficient algorithm that aims to find the optimal mapping. Experiment shows that our approach performs very well.

1 Introduction

We have witnessed an explosion in the amount of information that is available on the World Wide Web. A survey [42] in 1999 estimated a total of 800 million pages on the Web at that time. A recent index of Yahoo.com reported 19.2 billion Web pages, indicating an increase of about 24 times in six years. With such huge amount of information, the web has become the major information source in every corner of our daily life, including searching product and shopping information that has become a major part of our shopping process. However, as there are more and more online stores and shopping sites available on the web, it becomes very time-consuming, if not impossible, to visit and search all the shopping sites to find the best product to buy. There is great need to build an integrated shopping search system that provides a unified interface and metasearch capability to search and access all the shopping sites in one query submission.

Most of the online store and shopping sites, such as for example Yahoo! Shopping, PC-Home and Openfind, to name a few, provide category-constrained search in which a category is selected to reduce the search space, and improves both the response time and the quality of query results. To find products of interest, one usually navigates along the category directory to select a category, and then submits a query to find products in the selected category whose descriptions match the query. To build a system integrating category-constrained search, one need to solve the following fundamental problems.

- Category mapping that maps the selected category in the unified interface to categories provided by the information sources.
- Query translation that parses and matches query interfaces, and transform query predicates between interfaces.
- Wrapper induction that integrates the query results returned from information sources.

There are a lot of recent research on matching query interfaces[5, 6, 7, 14], transforming queries[19], and inducing wrappers[2, 3, 13] for result merging. In this paper, we focus on category mapping, and propose efficient approaches for finding mapping between category directories, such as, for example, category directories from shopping sites of Yahoo! and PC-Home. Note that category selection reduces the space to search, and improves the query time and quality of the query result. Quick and high quality response from each shopping site is very crucial to the performance of an integrated shopping search system. Our objective is to find a mapping that optimizes the performance of integrated search. In particular, we expect to find mappings that achieve high precision and recall while the query is restricted to mapped categories.

Given two category directories H_1 and H_2, we expect the mapping from H_1 to H_2 have the following property. Let h be a category in H_1 and $M(h)$ denote the mapping of h in H_2. Note that due to the heterogeneity between H_1 and H_2, the mapping from H_1 to H_2 can be a one-to-many mapping, and hence $M(h)$ can be set of categories in H_2. We expect that most of the instances in h belong to categories in $M(h)$, and most of instances returned from search under categories in $M(h)$ also belong to h. In this paper, we formulate the mapping problem as a search problem. Let $h \in H_1$ and $g \in H_2$ be two categories, $p = P(h|g)$ and $r = P(g|h)$. We use classification from machine learning to estimate p and r, and propose a greedy search to find a mapping that aims to maximize a measure derived from p and r. Experiment shows that our approach performs very well.

There are some related work [1, 4, 20] on ontology mapping, alignment and integration. Note that a category directory can be viewed as an ontology. However, most of the work on ontology mapping and integration aim to find all the subset/superset/related relations between every pair of classes from both ontologies. The discovered relations describe the relation between these two ontologies, and can be used as basis to integrate them. However, they did not discuss how to optimize query performance that is the main concern of this paper.

The remainder of this paper is organized as follows. Section 2 gives formulation of category mapping. Section 3 gives estimation of parameters. Section 4 presents approaches for searching optimal mappings. Section 5 gives compilation of data sets. Section 6 gives experimental results. Section 7 concludes and gives further remarks.

2 Formulation of Category Mapping

In this section, we define the category mapping problem to be solved in this paper.

Let H be a category directory that consists of a set of predefined categories. Let h be a category in H, and U denote the set of instances indexed and categorized by the information source. A query q constrained by category h returns a set $Q_H^U(q,h)$ of instances in U that is assigned to category h, and matches query q.

Let H_1 and H_2 be two category directories provided by two different information sources. Note that category directories are developed independently, are not identical and sometimes can differ a lot. Instances belonging to same category in H_1 may be distributed to several categories in H_2, and vice versa. Category mapping from H_1 to H_2 is often not a simple one-to-one mapping. We thus define category mapping as a function $M : H_1 \rightarrow 2^{H_2}$ that maps each category h in H_1 to a subset $M(h)$ of categories in H_2. In an ideal mapping, one may expect that, under the same instance set U, the same query q constrained by either h in H_1 or $M(h)$ in H_2 will return the same result, i.e. $Q_{H_1}^U(q,h) = Q_{H_2}^U(q,M(h))$, for all possible queries q under category h. Under above discussion, we can define the following ideal category mapping problem.

Let H_1 and H_2 be two category directories, U be the universal set of all possible data instances. The *ideal category mapping* problem is to find, for every $h \in H_1$, a set $M(h)$ of categories in H_2 so that $Q_{H_1}^U(q,h) = Q_{H_2}^U(q,M(h))$.

In practice, category directories are developed independently, and are heterogeneous. It is rare, if not impossible, to find an ideal mapping. Instead, one may want to find a mapping that minimizes the overall difference between $Q_{H_1}^U(q,h)$ and $Q_{H_2}^U(q,M(h))$, for all possible query q. Note that when most of the instances in h will be categorized to categories in $M(h)$ by H_2, and most of the instances in categories in $M(h)$ will be categorized to h by H_1, i.e. both conditional probabilities $P(M(h)|h)$ and $P(h|M(h))$ are high, the difference between $Q_{H_1}^U(q,h)$ and $Q_{H_2}^U(q,M(h))$ is expected to be small,for most of the query q under category h. In this paper, we propose to find a B that maximizes $P(B|h)$ and $P(h|B)$.

Instead of maximizing both $P(B|h)$ and $P(h|B)$ directly, it is common to define and maximize an objective function that combines $P(B|h)$ and $P(h|B)$. One of the common measures is the F_α-measure [10] that is a weighted version of harmonic mean as defined below. We define an objective function $F_\alpha(B,h) = \frac{(\alpha+1)P(B|h)P(h|B)}{P(B|h)+\alpha P(h|B)}$, where the weight of $P(h|B)$ is 1, and the weight of $P(B|h)$ is $\alpha \in (0,+\infty)$. We formulate the problem of category mapping as the following search problem.

Let $F_\alpha(B,h) = \frac{(\alpha+1)P(B|h)P(h|B)}{P(B|h)+\alpha P(h|B)}$, for every $h \in H_1$ and $B \subseteq H_2$. The *optimal category mapping* problem is to find, for every $h \in H_1$, a set $M(h)$ of categories in H_2 so that $M(h) = \arg\max_{B\subset H_2} F_\alpha(B,h)$.

Finding optimal mapping as formulated above faces the problem of probability estimation, and the problem of searching the optimal mapping, that will be discussed in following sections.

3 Estimation of Parameters

For every category $h \in H_1$ and subset $B \subset H_2$, we need to estimate $P(B|h)$ and $P(h|B)$ so that we can compute $F_\alpha(B,h)$ to find the optimal mapping of h. Notice that the number of possible combinations of h and B is $O(n2^m)$, where n and m are the number of categories in H_1 and H_2, respectively. Direct estimation of $P(B|h)$ and $P(h|B)$ for every possible combination of h and B is impractical. To simplify the problem, we assume that categories in H_2 are pairwise disjoint. In particular, when the original directory is hierarchical, we assume H_2 consists of the set of leaf categories in the original hierarchy, that are disjoint in general. Under the disjoint assumption, we can derive the following formula to compute $P(B|h)$ and $P(h|B)$.

From fundamental probability theories, we can derive the following lemma.

Lemma 1 $P(h|B) = \frac{\sum_{g\in B} P(h|g)P(g)}{\sum_{g\in B} P(g)}$

It is sufficient to estimate $P(h|g)$, $P(g|h)$ and $P(g)$, for every $h \in H_1$ and $g \in H_2$. This reduces the number of parameter estimation from $O(n2^m)$ to $O(nm)$. Note that $P(g)$ is estimated as the portion of instances in U_2 that belongs to category g. We next explain how to estimate above probabilities $P(h|g)$, $P(g|h)$.

WLOG, let $H_1 = \{h_1, h_2, \ldots, h_n\}$, and $H_2 = \{g_1, g_2, \ldots, g_m\}$, where categories in H_1, resp. H_2, are pairwise disjoint. Let U_1 and U_2 be the sets of instances indexed and categorized by H_1 and H_2, respectively. We use machine learning to estimate $P(g_j|h_i)$ and $P(h_i|g_j)$, for all $1 \le i \le n$ and $1 \le j \le m$. We use machine learning to learn a classifier C_1 from labelled data U_1, and C_2 from U_2. We then use classifier C_1 to classify all the instances in U_2, and produce a matrix $(a)_{n\times m}$ with a_{ij} denote the number of instances in category g_j that are classified to category h_i by C_1. Similarly, use classifier C_2 to classify all the instances in U_1, and produce a matrix $(b)_{n\times m}$ with b_{ij} denote the number of instances in category h_i that are classified to category g_j by C_2. We use matrices $(a)_{n\times m}$ and $(b)_{n\times m}$ to estimate probabilities $P(g_j|h_i)$ and $P(h_i|g_j)$ as follows: $\hat{P}(g_j|h_i) = b_{i,j}/\sum_{j=1}^{m} b_{i,j}$, and $\hat{P}(h_i|g_j) = a_{i,j}/\sum_{j=1}^{m} a_{i,j}$.

In the experiment of this paper, we simply use linear classifier, to estimate conditional probabilities. Nonetheless, any other classifier can be used as well. In the linear classifier, we compute a prototype for each class, which is the average of the instances in that class. To classify a new instance, we compute cosine similarity between it and every prototype, and assign to it the class whose prototype yields the largest similarity.

4 Searching the Optimal Mapping

Consider a category $h \in H_1$, and its optimal mapping $M(h) \subset H_2$ that maximizes F_α value. Exhaustive search that examines all possible mappings can find the optimal solution, but is impractical as it will take running time exponential to the number of categories in H_2 that can consists of hundreds of categories in applications such as shopping search. We propose a greedy approach that iteratively grows a category set and aims to find the optimal mapping. Experiment shows that our approach performs quite in real applications.

Let $B_i = \arg\max_{B\subset H_2, |B|=i} F_\alpha(B, h)$, $i = 1, \ldots, m$, i.e. B_i is the one that maximizes the F_α value among all the category sets of size i. Thus, $M(h) = \arg\max_{B_i, i=1,\ldots,m} F_\alpha(B_i, h)$. In the exhaustive search, for each i, we will generate all possible subsets of size i, compute their F_α value, and find B_i. This will take running time exponential to the number of categories in H_2 that can be as large as several hundreds in shopping applications. We propose a greedy subset growing approach to find a good mapping that performs quite well in our experiment.

In the greedy approach, we compute $\hat{B}_i$ that aims to approximate B_i, by examining the subsets that are extended from $\hat{B}_{i-1}$, instead of all subsets of size i. Let $\hat{B}_0 = \emptyset$. $\hat{B}_i$ is derived from $\hat{B_{i-1}}$ by adding a category g that maximizes $F_\alpha(\hat{B}_{i-1} \cup \{g\}, h)$ among all $g \in H_2 - \hat{B}_{i-1}$, i.e. $g = \arg\max_{g\in H_2 - \hat{B}_{i-1}} F_\alpha(\hat{B}_{i-1} \cup \{g\}, h)$.

5 Compilation of Data Sets

To evaluate the performance of our approach, we compile two data sets: one Chinese data set from web shopping sites in Taiwan, and the other English data set from web shopping sites in USA. For each web site, we download its web pages, using crawler HTTrack, to a local directory, and store them as HTML files. Note that category index as shown in Figure **??**that describes the category of a product is common in web pages downloaded from shopping sites. We identify HTML patterns for category index, and use pattern rules to recognize and extract category index from each web page. Each category index is a sequence of categories. We use category index to organize them into a hierarchial directory.

In Chinese data set, we collect web pages from shopping sites of Yahoo! and PChome in Taiwan. Table 7 gives number of pages and categories from each web site. In English data set, we collect web pages from web sites SuperPage, Shopping.com and shopLocal. Table 7 gives number of pages and categories from each web site. It is observed from both data sets that category directories from different web sites can be quite different. They may adopt difference names for the same categories, and different structures to organize categories.

For each web page, we run term extraction to extract terms from each page, and term selection to select informative terms. We use χ^2-statistic approach for term selection, that achieves the best overall performance as shown in [16]. Terms are weighted by TF/IDF scheme that is widely used in information retrieval.

Let n be the total number of selected terms. Each document d_j is represented as a vector $\vec{d_j} = (w_{1j}, w_{2j}, ..., w_{nj})$, where w_{ij} is the TF/IDF weight of term t_i in document d_j. Cosine similarity [?] is a widely used measure to determine the similarity between two documents that is defined as the inner product normalized by vector lengths. Given two vectors of documents, $\vec{d_j}$ and $\vec{d_k}$, the cosine similarity is defined as follows.

We use linear classifier for probability estimation. In linear classifier, each category is represented as a term vector. To classify a new document x, we compute the cosine similarity between x and every category, and assign to it the category that has the largest similarity. In our experiment, each category is represented as a term vector that is the sum of the term vectors of all documents under that category.

6 Experimental Result

In the experiment, we choose $\alpha = 1$ that means equal weight is assigned to $P(B|h)$ and $P(h|B)$. We implement a name-based approach in which two categories are mapped if they have the same name, and combine it with our approach.

In Chinese data set, only 20 mappings are identified by name-based approach. Our approach identify 67 mappings, and the combined approach is able to identify 74 mappings.Figure 1 gives some example of the mappings identified by our approach. Many categories, such as, for example, categories watches, pets and toys, in Yahoo! shopping are distributed to several categories in PChome. Our approach are able to discover them.

In English data set, similar result is achieved.

The experiment shows that our approach performs quite well.

7 Conclusion

In this paper, we present an efficient approach for mapping categories between directories from shopping web sites. We formulate category mapping as a search problem, and develop a greedy selection algorithm that is able to identify simple as well as complex mappings. Experiment shows that our algorithm perform quite well in both Chinese and English data sets. Note that category-constrained search is a very popular mechanism provided by many information sources. Although, in this paper, we focus on on applications in web shopping search, our approaches can be used to mapping categories in other applications such as online news as well.

$$Sim(d_j, d_k) = \cos(\vec{d_j}, \vec{d_k}) = \frac{\vec{d_j} \bullet \vec{d_k}}{|\ \vec{d_j}\ ||\ \vec{d_k}\ |} = \frac{\sum_{i=1}^{t} (w_{ij} * w_{ik})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 * \sum_{i=1}^{t} w_{ik}^2}} \quad (1)$$

Website name	URL	Number of pages	Number of categories
Yahoo!	http://buy.yahoo.com.tw/	6863	80
PChome	http://shopping.pchome.com.tw/	3738	144

Table 1: Data Source of Chinese Shopping Websites

Website name	URL	Number of pages	Number of categores
SuperPage	http://www.superpages.com/	22929	112
Shopping.com	http://www.shopping.com/	31922	110

Table 2: Data Source of English Shopping Websites

References

[1] Namyoun Choi, Il-Yeol Song, and Hyoil Han, A Survey on Ontology Mapping, College of Information Science and Technology Drexel University.

[2] S.-L. Chuang, K. C.-C. Chang, and C. Zhai. Collaborative Wrapping: A Turbo Framework for Web Data Extraction. In Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE), Istanbul, Turkey, April 2007.

[3] S.-L. Chuang, K. C.-C. Chang, and C. Zhai. Context-Aware Wrapping: Synchronized Data Extraction. To appear in Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB), Vienna, Austria, September 23-28 2007.

[4] AnHai Doan, Jayant Madhavan, Pedro Domingos, Alon Halevy, Learning to Map between Ontologies on the Semantic Web, VLDB Journal, Special Issue on the Semantic Web, 2003.

[5] Bin He, Kevin Chen-Chuan Chang. *Statistical Schema Matching across Web Query Interfaces.* In Proceedings of the 2003 ACM SIGMOD Conference (SIGMOD 2003), San Diego, California, June 2003.

[6] Bin He, Kevin Chen-Chuan Chang, Jiawei Han. *Discovering Complex Matchings across Web Query Interfaces: A Correlation Mining Approach.* In Proceedings of the 2004 ACM SIGKDD Conference (KDD 2004), Seattle, Washington, August 2004.

[7] Bin He, Kevin Chen-Chuan Chang. *Automatic Complex Schema Matching across Web Query Interfaces: A Correlation Mining Approach.* ACM Transactions on Database Systems (TODS), 31(1), March 2006.

[8] C. J. van Rijsbergen. *Information Retireval.* Butterworths, London, 1979.

[9] Baeza-Yates, Ricardo and Castillo, Carlos. *Crawling the infinite Web: five levels are enough.* In Proceedings of WAW, LNCS 3243, pp. 156-167. Rome, Italy, 2004. Springer.

[10] J.D.M. Rennie. *Derivation of the F-measure.* http://people.csail.mit.edu/jrennie/writing/, 2004

[11] Ching-Liang Kang. *Design and Development of an Integrated Product Search System.* Master's thesis, Department of Computer Science and Information Engineering, National Chung Cheng University, 2006.

[12] Chi-Hsiang Lin. *Time-Efficient Text Categorization for Web Directories.* Master's thesis, Department of Computer Science and Information Engineering, National Chung Cheng University, 2005.

[13] N. Kushmerick, D. S. Weld, and R. B. Doorenbos. Wrapper induction for information extraction. In Proc. of IJCAI, 1997.

[14] Weifeng Su, Jiying Wang, and Frederick Lochovsky. *Holistic Schema Matching for Web Query Interface.* In Proceedings of EDBT, 2006.

[15] Weifeng Su, Jiying Wang, Fred Lochovsky. *Automatic Hierarchical Classification of Structured Deep Web Databases.* The 7th International Conference on Web Information Systems Engineering (WISE), 2006.

[16] Jyh-Jong Tsay, Jing-Doo Wang. *Term Selection with Distributional Clustering for Chinese Text Classification using N-grams.* ROCLING XII, pages 151-170, 1999.

[17] Jiying Wang, Ji-Rong Wen, Frederick Lochovsky, Wei-Ying Ma. *Instance-based Schema Matching for Web Databases by Domain-specific Query Probing.* The 30th International Conference on Very Large Data Bases (VLDB 2004), Toronto, Ontario, Canada, August 2004.

[18] Yiming Yang, Jan O. Pedersen. *A comparative study on feature selection in text categorization.* In Proceedings of the Fourteenth International Conference on Machine Learing(ICML), 1997.

[19] Z. Zhang, B. He, and K. C.-C. Chang. Z. Zhang, B. He, and K. C.-C. Chang. On-the-fly Constraint Mapping across Web Query Interfaces. In Proceedings of the VLDB Workshop on Information Integration on the Web (VLDB-IIWeb'04), Toronto, Canada, August 2004.

[20] Naijun Zhou, A Study on Automatic Ontology Mapping of Categorical Information , Department of Geography, Land Information and Computer Graphic Facility University of Wisconsin V Madison.

	Yahoo! Shopping website	PChome shopping website	F_1-*measure* value
1	購物中心首頁/電腦資訊/LCD/	PChome/線上購物/零組件/LCD/	0.7741935483870968
2	購物中心首頁/鞋包配飾/手錶/	PChome/線上購物/男時尚/手錶大集合/ PChome/線上購物/男時尚/精品錶/ PChome/線上購物/男時尚/機械錶/	0.8409090909090908
3	購物中心首頁/鞋包配飾/飾品配件/	PChome/線上購物/女時尚/配飾/ PChome/線上購物/女時尚/名牌飾品大集合/ PChome/線上購物/女時尚/開運/	0.8555850532385599
4	購物中心首頁/鞋包配飾/休閒運動鞋/	PChome/線上購物/男時尚/男鞋/ PChome/線上購物/育樂/單品/	0.45454545454545453
5	購物中心首頁/書籍 DVD/文學小說/	PChome/線上購物/女時尚/精品大集合/	0.010526315789473684
6	購物中心首頁/居家生活/寵物/	PChome/線上購物/生活/寶貝訓練用品/ PChome/線上購物/生活/食用潔牙骨/ PChome/線上購物/生活/寵愛一世/ PChome/線上購物/生活/珍藏衣物/	0.8
7	購物中心首頁/居家生活/品牌寢具/	PChome/線上購物/生活/專櫃寢具/	0.6666666666666666
8	購物中心首頁/美妝/身體清潔/	PChome/線上購物/女時尚/美髮/ PChome/線上購物/女時尚/歐舒丹/ PChome/線上購物/女時尚/Burts/	0.6870020964360587
9	購物中心首頁/電腦資訊/PC/	PChome/線上購物/3C/DIY/ PChome/線上購物/3C/桌上電腦/	0.9049098819142324
10	購物中心首頁/消費電子/數位相機/	PChome/線上購物/3C/數位相機/	0.8702531645569619
11	購物中心首頁/書籍 DVD/DVD/	PChome/線上購物/育樂/DVD/	0.011695906432748537
12	購物中心首頁/超商取貨/美容保養/	PChome/線上購物/女時尚/歐舒丹/ PChome/線上購物/女時尚/KOSE/ PChome/線上購物/女時尚/流行女裝/	0.11834319526627218
13	購物中心首頁/書籍 DVD/健康美容/	PChome/線上購物/女時尚/男保養/ PChome/線上購物/女時尚/草本/	0.011374407582938386
14	購物中心首頁/消費電子/手機/	PChome/線上購物/3C/手機/	0.9670295730059455
15	購物中心首頁/書籍 DVD/雜誌/	PChome/線上購物/生活/兒童天地/	0.22058823529411764
16	購物中心首頁/電腦資訊/週邊/	PChome/線上購物/3C/電腦周邊/ PChome/線上購物/零組件/散熱精品/	0.6202723146747353
17	購物中心首頁/居家生活/玩具/	PChome/線上購物/育樂/玩具/ PChome/線上購物/生活/兒童天地/ PChome/線上購物/生活/存錢筒/	0.5084745762711864
18	購物中心首頁/品牌旗艦/Ocean/	PChome/線上購物/零組件/燒錄器/	0.5
19	購物中心首頁/美妝/香水/	PChome/線上購物/女時尚/香水/	0.961038961038961
20	購物中心首頁/居家生活/寢具/	PChome/線上購物/生活/專櫃寢具/	0.5714285714285715
21	購物中心首頁/視聽家電/廚房家電/	PChome/線上購物/家電/廚房家電/ PChome/線上購物/男時尚/精品男裝/	0.8217522658610271
22	購物中心首頁/超商取貨/服裝飾品/	PChome/線上購物/女時尚/名牌飾品大集合/	0.06666666666666667

Figure 1: Detailed mapping result of Yahoo! shopping website and PChome shopping website(1).