Mining Web Query Hierarchies from Clickthrough Data*

Dou Shen[1], Min Qin[2], Weizhu Chen[3], Qiang Yang[1], Zheng Chen[3]
[1]The Hong Kong University of Science and Technology, Hong Kong
{dshen, qyang}@cse.ust.hk
[2]Xi'an JiaoTong University, Xi'an, China
{ecogiser}@hotmail.com
[3]Microsoft Research Asia, 49 Zhichun Road, China
{wzchen, zhengc}@microsoft.com

Abstract

In this paper, we propose to mine query hierarchies from clickthrough data, which is within the larger area of automatic acquisition of knowledge from the Web. When a user submits a query to a search engine and clicks on the returned Web pages, the user's understanding of the query as well as its relation to the Web pages is encoded in the clickthrough data. With millions of queries being submitted to search engines every day, it is both important and beneficial to mine the knowledge hidden in the queries and their intended Web pages. We can use this information in various ways, such as providing query suggestions and organizing the queries. In this paper, we plan to exploit the knowledge hidden in clickthrough logs by constructing query hierarchies, which can reflect the relationship among queries. Our proposed method consists of two stages: generating candidate queries and determining "generalization/specialization" relations between these queries in a hierarchy. We test our method on some labeled data sets and illustrate the effectiveness of our proposed solution empirically.

Introduction

With the exponentially increasing amount of information being made available over the Internet, Web search is the most indispensable tool for Web users to gain their desired information. Typically, Web users submit a short search query consisting of a few words to some search engines, and obtain a list of search results in terms of Web pages located online. These queries, returned pages and subsequent user clicks on the pages, constitute clickthrough logs. Often, there is rich knowledge hidden in these logs. For example, by using the logs, one can draw a relationship among queries, and between queries and Web pages. Such a knowledge base can be considered as a hierarchical taxonomy, which allows one to specify the parent-child relationship between concepts and query terms. In this paper, we will explore how to mine query clickthrough logs in order to obtain a query hierarchy automatically.

One application of the mined knowledge hierarchy from clickthrough logs is to expand these queries using the query relationship given by the taxonomy. Because these queries are short and may be ambiguous, it has been a major research issue how to understand the queries and organize them well. In the past, many researchers have attempted to understand Web queries by expanding them through relevance feedback based on explicit interaction with Web users, or using pseudo-relevance feedback using the returned Web pages for the original query (Manning, Raghavan, & Schüze 2007). These methods however, have their shortcomings, because Web users are often reluctant to make extra effort to provide explicit feedback. However, if we know the relationship among the Web queries, we can refine Web users' queries more precisely and retrieve correct Web pages. We can also additionally provide more relevant query suggestions for Web search.

In this paper, we describe a novel knowledge discovery method for mining the "generalization/specialization" relations among Web queries and further build query hierarchies from the clickthrough data. As observed in (Fotzo & Gallinari 2004), the "generalization/specialization" relation can be understood as "is-a" relations. They are very intuitive for human users and widely used in many areas, such as online dictionaries. Figure 1 shows a fragment of a query hierarchy that we generate using our proposed method starting from the query "bmw".

Our work in building query hierarchies is related to previous works on building taxonomy and concept hierarchies (Buitelaar, Olejnik, & Sintek 2004; Cimiano, Hotho, & Staab 2005; Sanderson & Croft 1999). However, most of the previous works rely on a collection of documents or Web pages. They use either pattern-based approaches or distributional hypothesis-based approaches. However, these approaches cannot be easily applied to build the query hierarchies from clickthrough data, due to the *sparsity* and *noise* that are prevalent in Web queries and query logs. Therefore, in our method, we do not rely on the queries directly. Instead, we have to estimate the "generalization/specialization" relations between queries based on the user-clicked Web pages.

We make three basic assumptions in our algorithm for mining "generalization/specialization" relations . Firstly, if two queries are related to each other, they usually share the same or similar clicked Web pages. With this assumption, we can constrain the candidate set of queries when we find

*This work was conducted when Dou Shen and Min Qin were visiting Microsoft Research Asia, Beijing, China.

Copyright © 2007, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

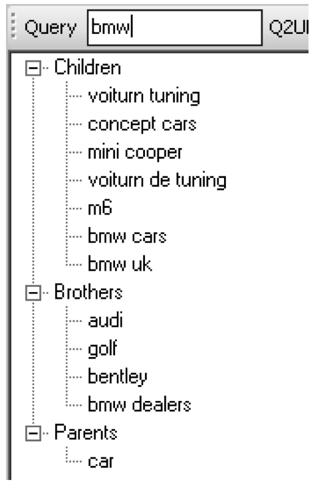

Figure 1: A fragment of the hierarchy for the query "bmw".

the parents or children of a given query. Our second assumption for deriving query hierarchies is similar to the idea of *term subsumption* presented by Sanderson and Croft in (Sanderson & Croft 1999). The term subsumption means that for two terms t_1 and t_2, t_1 subsumes t_2 if the documents in which t_2 occurs are a subset of the documents in which t_1 occurs. In this paper, we assume that for two queries q_i and q_j, q_i subsumes q_j if most of the clicked pages of q_j have similar pages in the clicked pages of q_i, while only part of the clicked pages of q_i have similar pages in the clicked pages of q_j. A third assumption of our method is that if a query is specific, the content of subsequent user-clicked Web pages are relatively consistent. Based on these assumptions, query hierarchy can be built into two steps: generating query candidates for a certain query, and determining the "generalization/specialization" relations among the queries.

We test our proposed method over a clickthrough data set which is sampled from a real search engine. We then ask human experts to judge the results. Our experimental results help validate the effectiveness of our methods.

In the rest of the paper, we first introduce some related works and then present our method. After that we show some experimental results on some real data sets. Finally, we conclude our work and point out possible directions of our future work.

Related Work

Our work in this paper is related to two major groups of work, including ontologies/concept-hierarchies construction and query analysis.

Taxonomies or concept hierarchies are widely used in different domains including information retrieval (Voorhees 1994), text clustering/classification (Bloehdorn, Cimiano, & Hotho March 9 11 2005; Bloehdorn & Hotho 2004) and some Natural Language Processing problems. However, it is not trivial to construct the taxonomies and concept hierarchies. Traditionally, researchers have created some taxonomies manually, such as WordNet (Miller *et al.* 1990), which is quite time-consuming and hard to update. Recently, people have started working on automatic approaches for building taxonomies. There are currently three main paradigms exploited to induce taxonomies from textual data (Buitelaar, Cimiano, & Magnini 2005). The first one is the application of lexico-syntactic patterns to detect hyponymy relations as proposed by (Hearst 1992). However, it is true that these patterns occur rarely in corpora. Thus, the approaches relying on lexico-syntactic patterns feature a reasonable high precision and low recall. Related to this are also approaches that exploit the internal structure of noun phrases to derive taxonomic relations between classes expressed by the head of the noun phrase and its subclasses that can be derived from a combination of the head and its modifiers (Buitelaar, Olejnik, & Sintek 2004). The second paradigm is based on Harris' distributional hypothesis claiming that terms are semantically similar to the extent to which they share similar syntactic contexts (Harris 1968). Along this line, people have mainly exploited hierarchical clustering algorithms to automatically derive term hierarchies from text (Cimiano, Hotho, & Staab 2005). The third paradigm stems from the information retrieval community and relies on a document-based notion of term subsumption as proposed for example in (Sanderson & Croft 1999). For two terms t_1 and t_2, these method assume that t_1 subsumes t_2 if the documents in which t_2 occurs are a subset of the documents in which t_1 occurs.

The problem in this paper is similar to the problem of building taxonomies and concept hierarchies in that we hope to build query hierarchies in which the pairs of queries have the "generalization/specialization" relations. However, our problem is clearly different from the previous work on building taxonomies since we are working on clickthrough data where the Web queries are quite short and noisy. However, the clickthrough data incorporate human knowledge of the queries as well as their relations to the clicked pages. Exploiting human knowledge as reflected via clickthrough data to estimate the relations among queries accurately is the focus of this paper.

Since we are working on Web queries in clickthrough data, our problem is also related to some work about query analysis. In the past, researchers have worked along several different lines to understand Web queries. Firstly, query classification have been widely studied to uncover the semantic nature of query terms. In query classification, Gravano et al. discuss how to categorize queries according to their geographical locality (Gravano, Hatzivassiloglou, & Lichtenstein 2003). Kang et al. (Kang & Kim 2003) classify the queries into two service-type categories based on their goals: topic relevance task (informational) or homepage finding task (navigational). Lee et al. further study whether the type of query is predictable and how to predict it (Lee, Liu, & Cho 2005). Pu et al. use a classifier built using manual labeling and semi-supervised mining of retrieved web pages to categorize frequent query terms in 15 major categories and 85 subcategories (Pu, Chuang, & Yang 2002). In (Shen *et al.* 2006), we studied the topic classification of Web queries, where the Web queries are classified into some predefined categories based on their topics without training data for predefined categories.

Besides query classification, query clustering provides an alternative way to understand Web queries. Beeferman and Berger mined a log of 500,000 clickthrough records from the Lycos^{TM} search engine and used a bipartite-graph algorithm to discover latent relationships between queries based on common click-through documents linking them together (Beeferman & Berger 2000). They make no use of the actual content of the queries and URLs, which is then "content-ignorant". Conversely, Wen et al. represent the queries by their content and clicked documents in the query/click logs (Wen, Nie, & Zhang 2001). They concluded that the use of query keywords together with session data is the most effective method of performing query clustering. However, their test collection was an encyclopedia, so the applicability of their results to general web search is limited.

In this paper, instead of working on query classification and clustering, we analyze the Web queries from a new perspective by building query hierarchies. As far as we know, this paper is the first attempt to work on query hierarchies.

Sharing the same spirit with our work, some researchers try to take the clickthrough data as a knowledge base and mine valuable information from it. In (Paşca & Durme 2007), the authors introduce a method for extracting attributes, or quantifiable properties, for various classes of objects. Different from previous approaches to large-scale information extraction, they explore the role of Web query logs, rather than Web documents, as an alternative source of class attributes. The quality of the extracted attributes recommends query logs as a valuable, albeit little explored, resource for information extraction.

Our Solution

In this paper, we focus on finding the hierarchical structure around a given query in a query clickthrough log. Given a query q, we find out its parents/ancestors, children/descendants and brothers. Here we do not distinguish between parents (children) and ancestors (descendants). Actually, it is not hard to put the hierarchical structures of single queries together to form an overall hierarchical structure among all queries. However, local structures around single queries are enough in most scenarios including query expansion and suggestion. What is more, as claimed in (Sanderson & Croft 1999), it may be meaningless to transit parent-child relations. Using an example from (Sanderson & Croft 1999), a "ships captain" is a "profession" and "Captain Ahab" is a "ships captain", but the relationship between "Captain Ahab" and the concept "profession" is less clear. Therefore, we focus on local structures around single queries.

To derive the local hierarchical structure of a given Web query, we make three assumptions based on our observations on the clickthrough data collected from Live Search [1].

- If two queries are related to each other, they should share some of the same or similar clicked Web pages;
- For two queries q_i and q_j, q_i is q_j's parent if most of the clicked pages of q_j have similar pages to the clicked pages of q_i while only part of the clicked pages of q_i have similar pages to the clicked pages of q_j;
- If a query is specific, the contents of its clicked pages are relatively consistent, compared to a general query.

[1] http://www.live.com

Based on assumption 1, we can limit the candidate set for a given query q effectively. That is, we just build the local structure of q by considering the queries sharing common clicked pages with q. With assumption 2, we can further reduce the size of the candidate set by removing the queries whose clicked pages are not similar to the clicked pages of q. We can further determine the relative generality and specificity between two terms using assumption 2. Finally, we can use the third assumption to refine the estimation of the "generalization/specialization" relations between two terms. Before discussing the two steps for generating queries hierarchies in detail, let us formally define the measurements in assumption 2 and 3 by introducing *relative coverage* and *specificity*.

Relative Coverage (RC) Given two queries q_i and q_j, assuming q_i is q_j's parent, the above assumption 2 means that most of the clicked pages of q_j are covered by the clicked pages of q_i while just part of the clicked pages of q_i are covered by those of q_j. Therefore, we introduce Relative Coverage ($RC(q_i, q_j)$) to measure the extent to which the clicked pages of one query (q_i) is covered by the clicked pages of another query (q_j). $RC(q_i, q_j)$ is defined by equation (1).

$$RC(q_i, q_j) = \frac{\sum_{n=1..N} \max_{m=1..M} sim(d_{in}, d_{jm})}{N} \qquad (1)$$

In equation (1), $d_{in} \in D(q_i)$ and $d_{jm} \in D(q_j)$ where $D(q)$ means the set of clicked pages of query q. N and M are the size of $D(q_i)$ and $D(q_j)$. $sim(d_{in}, d_{jm})$ measures the similarity between d_{in} and d_{jm} which is generally cosine similarity. We can use different ways to represent the clicked pages. Take d_{in} as an example.

- We can use all the text in d_{in} after removing the HTML tags (denoted by *Full Text* in the experiments);
- We can use the snippet of d_{in} generated by search engines for the query q_i. The snippet is usually a query-dependent summary of the page (denoted by *Snippet*);
- We can use the union of the queries which are related to d_{in} in the clickthrough data, which equals to $\{q_k | d_{in} \in D(q_k)\}$ (denoted by *Queries*).

We study the performance of different ways for representing clicked pages in the experiments.

It is clear that if most of clicked pages of q_i have similar pages among the clicked pages of q_j, $RC(q_i, q_j)$ is large, which accords with assumption 2.

Specificity (Spec) In order to measure the consistency among the clicked pages of a given query q_i, which can reflect q_i's specificity, we introduce the measurement Specificity (Spec). The specificity of a query q_i is defined by equation (2).

$$Spec(q_i) = \frac{\sum_{n=1..N} \sum_{m=1..N; m \neq n} sim(d_{in}, d_{im})}{N(N-1)} \quad (2)$$

Now we introduce the two steps to build local hierarchical structure of a given query q_i: generating candidates and determining "generalization/specialization" relations.

Generating Candidates

Given the query q_i, we can follow the first assumption to find out all the queries in the clickthrough data which share clicked pages with q_i. This can greatly reduce the size of the potential candidates compared to the ways in previous work on building taxonomies. However, due to the noise in clickthrough data, we may still have some queries which are loosely related to q_i. Therefore, we can further reduce the candidate query set by implying the following requirement over any candidate query q_j:

$$RC(q_i, q_j) > \theta_1 \ \ and \ \ RC(q_j, q_i) > \theta_1 \quad (3)$$

The above requirement implies that for two queries to be related, they should have a certain amount of clicked pages which have similar pages in another query's clicked pages. The amount should be large enough to remove the possibility of introducing noise.

Determining "Generalization/Specialization"

Following assumption 2, we can estimate the relative generality and specificity between two queries q_i and q_j based on $RC(q_i, q_j)$ and $RC(q_j, q_i)$. The rule is as follows:

$$q_i \ is \ q_j's \ \ parent \ if \ RC(q_j, q_i) - RC(q_i, q_j) \ > \ \theta_2 \quad (4.1)$$

$$q_i \ is \ q_j's \ child \ if \ RC(q_i, q_j) - RC(q_j, q_i) \ > \ \theta_2 \quad (4.2)$$

$$Otherwise \ q_i \ and \ q_j \ are \ brothers \quad (4.3)$$

Since the specificity scores reflect queries' specificity, we can refine the RC score by adding the specificity scores when comparing the relative specificity of two queries. That is, we change $RC(q_i, q_j)$ and $RC(q_j, q_i)$ in the above rules to $\frac{RC(q_i,q_j)+Spec(q_i)}{2}$ and $\frac{RC(q_j,q_i)+Spec(q_j)}{2}$ respectively.

Let us explain the idea of our method with the example shown in Figure 2. There are three queries in Figure 2. We use triangles to represent the topics of the clicked pages about "bmw" and circles to represent those about "audi". The topics of the clicked pages about "car" are mixtures of topics of "bmw", "audi" and others. Given two queries, one parent and one child, the child focuses on a more specific topic which is covered by the parent. In Figure 2, the query "bmw" is a specific brand of "car" while "car" may also mean other brands such as "audi". It is reasonable to assume that Web users may click on Web pages about "bmw" and "audi" with comparable probability when they issue the query "car" without any prior knowledge of the brands. The assumption becomes more accurate with more clickthrough data. It is also reasonable to assume that Web users will be more likely to click on Web pages about "bmw" than "audi"

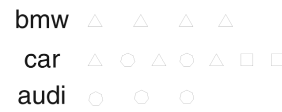

Figure 2: Illustrating example of our method.

Table 1: RC (R) and Spec (S) scores of several query pairs.

q_1	q_2	$R(q_1, q_2)$	$R(q_2, q_1)$	$S(q_1)$	$S(q_2)$
bmw	audi	0.395	0.392	0.196	0.228
bmw	car	0.505	0.360	0.196	0.140
audi	car	0.416	0.369	0.228	0.140

when they issue the query "bmw". Therefore, the clicked pages of "bmw" should be covered by the clicked pages of "car" but not vice versa. From Figure 2, we can easily understand that the specificity scores "bmw" and "audi" are higher than that of "car". We say two queries are brothers if they are similar to some extent as required by the parameter θ_1 in formula (3). However, none of them is much more specific than the other so that they cannot form the parent-child relation. In this paper, the brother of a query q may be q's uncle's child so long as they satisfy the formula (3) and (4.3).

Table 1 presents the RC and Spec scores for the three queries pairs from "car", "audi" and "bmw". From table 1, we can see that the scores verify our idea exactly. For example, $RC(bmw, car) > RC(car, bmw)$, $RC(audi, car) > RC(car, audi)$, $RC(bmw, audi) \approx RC(audi, bmw)$, $Spec(bmw) > Spec(car)$ and so on.

As we can see, there are two parameters in these two steps, θ_1 and θ_2. We study their roles empirically in the experiments.

Experiments

In this section, we test our proposed method over some human labeled data sets collected from a randomly sampled clickthrough data set of Live Search. We study the three assumptions as well as the impact of the several methods to represent clicked pages. The experimental results validate the effectiveness of our proposed method.

Performance of Generating Candidates

Note that the goal of the step of generating candidates is to remove unrelated queries given a certain query. To study the parameter θ_1 and the different ways to represent clicked pages, we randomly collected 280 pairs of unrelated queries and 200 pairs of related queries. Table 2 shows several examples of the pairs. Figure 3 shows the accuracy of our methods with different ways to represent clicked pages when we change the threshold. From Figure 3, we can see that when we use *queries* to represent clicked pages, our method achieves best performance when θ_1 equals to 0.03 and its performance is stable when we increase the parameter θ_1 to 0.30. The wide range of the parameter θ_1 for good performance makes it easier to tune the parameters in real applications. When we use *Full Text* and *Snippet* to represent clicked pages, the best performance is comparable to that achieved by *Queries*. However, the range of θ_1 is quite narrow. Therefore, we use *Queries* to represent clicked pages and set θ_1 to 0.10 in the consequent experiments for generating candidates.

Table 2: Examples of related/unrelated query pairs.

Unrelated Query Pairs	**Related Query Pairs**
<music, pet>	<music, video>
<mother day gift, pet store>	<car, bmw>
<top story, internet market>	<microsoft, windows xp>

Table 3: Number of clicked URLs of queries.

QUERY	music	bmw	xbox	car	windows xp
#URL	3571	4358	2428	8199	3013
QUERY	software	ea	game	movie	office 2007
#URL	1414	300	8866	5189	1750

Performance of Determining "Generalization/Specialization"

After testing our method of removing unrelated queries for generating candidates, we select 10 popular and easy-to-understand Web queries to test our method of determining "Generalization/Specialization" of query pairs. The ten queries are *music, bmw, windows xp, car, xbox, ea, software, game, movie, office 2007*. Table 3 shows the number of URLs associated with these queries. These numbers are obtained from the sampled clickthrough data of Live Search. For each of these 10 queries, we generate the candidate queries using the above method. For the convenience of labeling, we randomly sample 40 queries from the reasonable candidates of each query, which results in 10*40 query pairs. After that, we invite three labelers to label each query pair $< q_i, q_j >$ with three possible labels (q_i is q_j's parent; q_j is q_i's parent; q_i and q_j are brothers). We use voting when the three labelers are not consistent in the labeling results. Finally, we test our method to determine the "Generalization/Specialization" of the query pairs. Figure 4 shows the accuracy of determining "Generalization/Specialization" with different ways to represent clicked pages when we change the threshold θ_2.

In Figure 4, $Snippet$ means that we use the snippet to represent clicked pages. $Snippet^*$ means that we refine the RC score with the specificity score, where both scores are calculated by representing clicked pages by snippets. Similarly, we can explain *Full Text*, $Full\ Text^*$, $Queries$ and $Queries^*$. From Figure 4 we can see that by considering the specificity scores, we can improve the accuracy clearly except for $Snippet$ whose accuracy is even decreased. The reason is as follows. The snippets of the clicked pages given a certain query are quite similar to each other since they are actually a query-dependent summary of the corresponding pages. Therefore, the snippets are biased by the query and the specificity score calculated based on the snippets tend to be high, which can further overwhelm the RC scores. However, the specificity scores based on snippets cannot reflect the specificity of the query correctly.

Discussion

From the above experimental results, we can see that our method to generate candidates achieves promising performance while we still have room to improve the method of determining "Generalization/Specialization" relations. The reason is that the task of generating candidates is to remove unrelated query pairs which is relatively easy and determinable. However, the task of determining "Generalization/Specialization" relations is much harder because of the ambiguity of Web queries which makes the judgement obscure. For example, for the two queries "music" and "Yahoo!", we can say that "Yahoo!" is the parent of "music" in that "music" is one of Yahoo!'s channels which has other channels such as "Yahoo! sports", "Yahoo! financial". We can also say that "Yahoo!" is the child of "music" since "Yahoo!" is only one of the platforms providing online music. We leave it as our future work to find some approaches to handle the ambiguity problem of Web queries so that we can make the task of determining "Generalization/Specialization" relations easier.

Conclusion and Future Work

In this paper, we studied the problem of how to construct query hierarchies from the clickthrough log of a Web search engine. Our method can find the parents, children and brothers of Web queries by using the clickthrough data. We put forward two concepts "Relative Coverage" and "Specificity" which are shown to be useful in measuring the "Generalization/Specialization" relations between two queries. With these two concepts, we proposed a two-step method to build a local hierarchical structure of each query. The experimental results over some human labeled data sets validate the effectiveness of our methods.

In our future work, we plan to continue to verify our methods over some larger data sets. We also hope to find some more applications of the query hierarchies including query expansion and query suggestion.

References

Beeferman, D., and Berger, A. 2000. Agglomerative clustering of a search engine query log. In *KDD '00*, 407–416.

Bloehdorn, S., and Hotho, A. 2004. Text classification by boosting weak learners based on terms and concepts. In *ICDM '04*, 331–334.

Bloehdorn, S.; Cimiano, P.; and Hotho, A. March 9-11, 2005. Learning ontologies to improve text clustering and classification. In *GfKl '05*, 334–341.

Buitelaar, P.; Cimiano, P.; and Magnini, B., eds. 2005. *Ontology Learning from Text: Methods, Evaluation and Applications*, volume 123 of *Frontiers in Artificial Intelligence and Applications Series*. Amsterdam: IOS Press.

Buitelaar, P.; Olejnik, D.; and Sintek, M. 2004. A protégé plug-in for ontology extraction from text based on linguistic analysis. In *ESWS '04*.

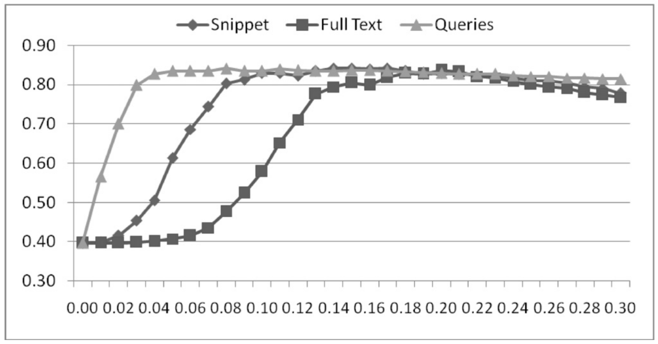

Figure 3: Accuracy (Y-axis) of generating candidates with different ways to represent clicked pages when we change the threshold θ_1 (X-axis).

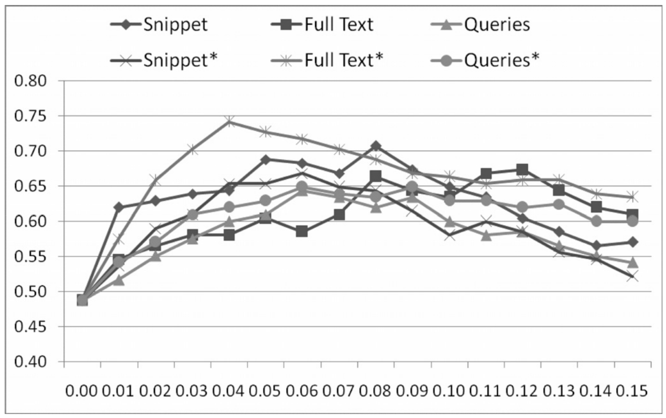

Figure 4: Accuracy (Y-axis) of determining "Generalization/Specialization" with different ways to represent clicked pages when we change the threshold θ_2 (X-axis).

Cimiano, P.; Hotho, A.; and Staab, S. 2005. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research (JAIR)* 24:305–339.

Fotzo, H. N., and Gallinari, P. 2004. Learning ≪ generalization/specialization≫ relations between concepts c application for automatically building thematic document hierarchies. In *RIAO '04*.

Gravano, L.; Hatzivassiloglou, V.; and Lichtenstein, R. 2003. Categorizing web queries according to geographical locality. In *CIKM '03*, 325–333.

Harris, Z. 1968. *Mathematical Structures of Language*. New York: Interscience Publishers John Wiley & Sons.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *ACL '92*, 539–545. Morristown, NJ, USA: Association for Computational Linguistics.

Kang, I.-H., and Kim, G. 2003. Query type classification for web document retrieval. In *SIGIR '03*, 64–71.

Lee, U.; Liu, Z.; and Cho, J. 2005. Automatic identification of user goals in web search. In *WWW '05*, 391–400.

Manning, C. D.; Raghavan, P.; and Schüze, H. 2007. *Introduction to Information Retrieval*. Cambridge University Press.

Miller, G.; Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, K. 1990. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography* 3(4):23–244.

Paşca, M., and Durme, B. V. 2007. What you seek is what you get: Extraction of class attributes from query logs. In *IJCAI '07*, 2832–2837.

Pu, H.-T.; Chuang, S.-L.; and Yang, C. 2002. Subject categorization of query terms for exploring web users' search interests. *J. Am. Soc. Inf. Sci. Technol.* 53(8):617–630.

Sanderson, M., and Croft, B. 1999. Deriving concept hierarchies from text. In *SIGIR '99*, 206–213.

Shen, D.; Pan, R.; Sun, J.-T.; Pan, J. J.; Wu, K.; Yin, J.; and Yang, Q. 2006. Query enrichment for web-query classification. *ACM Trans. Inf. Syst.* 24(3):320–352.

Voorhees, E. M. 1994. Query expansion using lexical-semantic relations. In *SIGIR '94*, 61–69.

Wen, J.-R.; Nie, J.-Y.; and Zhang, H.-J. 2001. Query clustering using content words and user feedback. In *SIGIR '01*, 442–443.