Expert Systems with Applications 39 (2012) 10739–10748

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A feature-free search query classification approach using semantic distance ☆

Lin Li [a,*], Luo Zhong [a], Guandong Xu [b], Masaru Kitsuregawa [c]

[a] School of Computer Science & Technology, Wuhan University of Technology, China
[b] Centre for Applied Informatics, Victoria University, Australia
[c] Institute of Industrial Science, University of Tokyo, Japan

ARTICLE INFO

Keywords:
Search query
Semantic distance
Page count
Classification

ABSTRACT

When classifying search queries into a set of target categories, machine learning based conventional approaches usually make use of external sources of information to obtain additional features for search queries and training data for target categories. Unfortunately, these approaches rely on large amount of training data for high classification precision. Moreover, they are known to suffer from inability to adapt to different target categories which may be caused by the dynamic changes observed in both Web topic taxonomy and Web content. In this paper, we propose a feature-free classification approach using semantic distance. We analyze queries and categories themselves and utilizes the number of Web pages containing both a query and a category as a semantic distance to determine their similarity. The most attractive feature of our approach is that it only utilizes the Web page counts estimated by a search engine to provide the search query classification with respectable accuracy. In addition, it can be easily adaptive to the changes in the target categories, since machine learning based approaches require extensive updating process, e.g., re-labeling outdated training data, re-training classifiers, to name a few, which is time consuming and high-cost. We conduct experimental study on the effectiveness of our approach using a set of rank measures and show that our approach performs competitively to some popular state-of-the-art solutions which, however, frequently use external sources and are inherently insufficient in flexibility.

1. Introduction

Web search engines today allow users to pose queries simply in terms of keywords. Most of the search queries are expressed by only a couple of keywords (Jansen, Spink, Bateman, & Saracevic, 1998). As a result, search queries may be ambiguous. This is because Web users choose terms to represent their information needs and search queries reflect implicit and subjective user intents. In fact, some queries have different meanings in different contexts. For example, "Python" can mean an animal or a programming language. It is very challenging for search engines to understand Web users' search goals only through their search queries and satisfy the users with high quality results. Search query classification is a well-known technique to help us understand user search intents, thus improving effectiveness and efficiency of Web search along several dimensions. In general Web search, search engines can organize the large number of Web pages in the search results according to the potential categories of the issued query, which provides the convenience for Web users to navigate the search results. Search query classification is also a basic component in constructing and utilizing user models to cater the search services to individual or a group of users' preferences. In addition, search engines can provide relevant advertisements to Web users according to their search interests, which has shown to significantly increase the effectiveness of online advertising.

1.1. Query classification vs text classification

Classification automatically assigns some predefined categories to objects, e.g., queries or texts. Most work in text classification and categorization has been focused on classifying static, feature-rich text corpus (Chen, Lee, & Chang, 2009; He, Duan, Zhou, & Dom, 2009; Kim, Pantel, Duan, & Gaffney, 2009; Özel, 2011). However, search query classification is very different in the sense that search queries are usually very short and ambiguous. Thus, the classification quality would be extremely unsatisfactory using the queries keywords alone. Moreover, the meanings of the search queries evolve over time reflecting the evolution of word usages in the real world society (Beitzel, Jensen, Chowdhury, Grossman, & Frieder, 2004). For example, after the late 1980s, the word "Python", which generally means a kind of large snakes, can also mean a kind of

☆ This research was undertaken as part of Project 61003130 funded by National Natural Science Foundation of China.

* Corresponding author.
E-mail addresses: cathylilin@whut.edu.cn (L. Li), zhongluo@whut.edu.cn (L. Zhong), guandong.xu@vu.edu.au (G. Xu), kitsure@tkl.iis.u-tokyo.ac.jp (M. Kitsuregawa).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.02.191

high-level programming languages. Therefore, the topical classification of search queries is much more difficult and challenging than traditional text classification tasks.

Most conventional approaches to search query classification are machine learning based. They rely on external sources of information, such as online Web directories (Shen, Yang, Sun, & Chen, 2006; Shen et al., 2006), the contents of retrieved pages (Pu et al., 2002; Shen, Yang, Sun, & Chen, 2006; Shen et al., 2006), query log data (Beitzel, Jensen, Lewis, Chowdhury, & Frieder, 2007; Pu et al., 2002) and so on, to obtain additional features for short queries and training data for target categories, which in turn facilitate the classification. A major drawback of these machine learning based approaches is that they cannot be easily adaptive to the changes of target categories which may be caused by the needs of service providers as well as the changes of the distribution of the Web topic taxonomy and Web content. Reflecting these changes to a machine learning based classification model is a time-consuming and high-cost process. It requires extensive updating in the system, e.g. re-labeling outdated training data, re-training classifiers, re-extracting queries features, to name a few. Therefore, automating the learning and adaptation process of the search query classification models becomes the big challenge, especially in the absence of up to date training data.

Shen et al. (2006) have started to address the flexibility problem. They first builds a bridging classifier on an intermediate taxonomy, such as Open Directory Project (ODP) in an offline phase. The bridging classifier is then used in an online phase to map user queries to any target categories with the aid of the contents of their retrieved Web pages. However, ODP, a manually maintained Web directory, cannot keep pace with the growth of the Web, thus having difficulty in covering a broad knowledge enough to deal with the changes of categories. Moreover, target categories can come from different domains, such as medicine, agriculture, literature, and so on. Using a same taxonomy for various kinds of categories is not fully flexible.

Bearing this problem in mind, we present a feature-free search query classification approach using semantic distance. Our approach makes three main contributions.

(i) We argue that it is possible to topically classify search queries without requiring manually labeling sufficient number of sample queries or gathering extra information as training data.
(ii) We propose to utilize the number of Web pages containing both a query and a category as a semantic distance to measure the similarity between a query and a category. As we know, the WWW is the largest database on earth, and the context information entered by millions of independent users averages out to provide automatic semantics of useful quality. The page counts returned by a search engine can be regarded as a compressor which encodes the updated semantics of a word or phrase in the context of the Web viewed by a search engine. The most attractive feature of our approach is that it only utilizes the Web page counts retrieved from a search engine, instead of performing text process on external sources. This makes our approach easily adaptive to the changes of target categories.
(iii) We study two classes of ranking strategies including six rank measures for classification of a query according to the given collection of categories. The rank measure computes query-category similarity score for every query-category pair and determines the best way of mapping a query to one or more of one or more categories. Our experimental results show that our approach is much better than the one that uses traditional WordNet based semantic distance measure and performs competitively to some popular state-of-the-art solutions which, however, rely on expensive external sources and are inherently insufficient in flexibility.

The rest of this paper is organized as follows. We first describe our approach in Section 2. The experiments results then are reported in Section 3. We discuss some related works in Section 4. Last, a summary of main contributions is given in Section 5 with some possible future research issues.

2. Our feature-free approach using semantic distance

In this section, we describe our approach for feature-free classification of search queries. First, we briefly summarize the problem statement and give an overview of our approach. Then, we discuss how to use the aggregate page-count estimates returned by a search engine to design a semantic ranking function which determines the categorization confidence of a query belonging to a category. Last, we analyze the time complexity of the approach.

2.1. Problem statement

search query classification refers to the capability of classifying a search query into one or several predefined target categories. In the case of large-scale Web search, this task is particularly challenging for a number of reasons:

- search queries are short and ambiguous. Traditional approaches for text classification, especially machine learning based methods do not work well if no additional features are obtained to enrich queries.
- To our best knowledge, most of the state of art approaches either need to re-train their classifiers or require heavy human labor to re-label queries whenever the target category structure changes.
- Web content and Web user behavior are highly dynamic. The information on the Web changes by about 8% per week (Beitzel et al., 2007), with content updating in servers and Web pages. Additionally, the Web users, their interests, their model of how a search works, and the needs of service providers are simultaneously and continuously varying, which leads the changes of target categories.

We argue that a classification system for search queries should be good at automatically classifying a large portion of the query stream with a reasonable degree of accuracy while it is flexible to new target categories.

2.2. Overview of our approach

We propose a novel feature-free approach, which can automatically and effectively classify search queries, and easily adapt to the changing category structures and a various types of target taxonomies when applied in different domains. In addition, our approach does not rely on any manually labeled training data, which are very expensive to obtain and maintain. Concretely, the main idea of our approach is twofold: First, we treat queries and category names as common words, and the probability of a query belonging to a category is converted to the similarity between two words or two bags of words. Second, we compose a search that combines a query and a category name and the similarity between them can be approximated based on their page counts (frequencies) in the index of a Web search engine.

The page-count is an estimate of the number of web pages where the query occur. Given a particular search query, say "apple", a Web search engine returns a certain page-count, say 480,000,000. It is also possible to search for web pages where two words occur together. For example, by submitting a query "apple + computer" to a Web search engine, we receive a page-count of 25,400,000. Yet our approach is flexible to a various types of target taxonomies when applied in different domains.

The overall process of our approach can be described as follows.

Step 1. We concatenate the input query and the category name as a whole search query, and submit it to a Web search engine to get a page-count.
Step 2. Their similarity is estimated by the page-count estimates. This process is repeated for all the target categories.
Step 3. The classification result is given as a rank list of categories ordered by a ranking strategy. The top N categories will be returned as the output to the user where N is a system supplied parameter determined by different system requirements.

Note that if the target categories are organized as a flat structure, we directly use the category names; if the target categories are organized as a hierarchical structure where each node represents a category, the labels along the path from the root to the nodes of the target categories are concatenated and used as the category names. For example, if the leaf node in the second level of a topical taxonomy is "Hardware" and its parent node is "Computers", the leaf node is represented by "Computers\Hardware".

2.3. Ranking strategies

Given a search query and the target categories, our approach computes the categorization confidences of the query belonging to the categories. In this section, we briefly discuss two classes of ranking strategies which assign scores on target categories for each query. The first strategy is a naïve ranking using maximum likelihood estimate, and the second strategy is a normalized ranking using five popular similarity measures. Before diving into the details of both ranking strategies, we give the terminologies used in the remaining discussion in Table 1.

2.3.1. Naïve ranking stragegy

The maximum likelihood estimate (MLE) of the co-occurrence probability is defined as:

$$R(q,c) = \mathrm{MLE}(q,c) = \frac{\log f(q,c)}{\log M}. \quad (1)$$

In computation $f(q,c)$ is added by one, i.e., $\log(f(q,c)+1)$ which makes sure that the variable used in log function is larger than zero. The parameter M is set to be a constant and usually its value is the total number of Web pages indexed by a search engine. The more Web pages found contain the two words q and c, the higher probability that q belongs to the category c.

However, the MLE rank measure is naïve, which utilizes the page counts $f(q,c)$ alone. For accurately expressing semantic similarity, one should also consider $f(q)$ and $f(c)$ (i.e., the number of Web pages containing q or c). In the following, we introduce five popular co-occurrence based similarity measures (Bollegala, Matsuo, & Ishizuka, 2007; Cilibrasi & Vitányi, 2007), which can be regarded as the normalized $f(q,c)$.

2.3.2. Normalized ranking strategy

The following five rank measures use different normalization functions for the page count $f(q,c)$. The normalization scales down the coordinate of (q,c) by considering the individual occurrences $f(q)$ and $f(c)$ because a word with high page frequency in a search engine might be less informative.

2.3.2.1. Normalized google distance. Cilibrasi and Vitányi (2007) presented a new theory of similarity between words based information distance and Kolmogorov complexity, and the new dissimilarity measure is called Normalized Google Distance (NGD) since it is based on the page counts returned by Google. We can view NGD as a way of utilizing $f(q,c)$ to compute the probability $R(q,c)$ that is defined as:

$$R(q,c) = \frac{1}{\mathrm{NGD}(q,c)} = \frac{\log M - \min\{\log f(q), \log f(c)\}}{\max\{\log f(q), \log f(c)\} - \log f(q,c)}. \quad (2)$$

Intuitively, $\mathrm{NGD}(q,c)$ is a measure for the symmetric conditional probability of co-occurrence of the query q and the category c over a collection of Web pages. $\mathrm{NGD}(q,c)$ computes a score that estimate the probability that the query q belongs to the category c. Although NGD is called normalized Google distance, we can get page counts from any available search engines.

2.3.2.2. Jaccard. The Jaccard rank measure is defined as:

$$R(q,c) = \mathrm{Jaccard}(q,c) = \frac{\log f(q,c)}{\log f(q) + \log f(c) - \log f(q,c)}. \quad (3)$$

Eq. (3) is a modified version of Jaccard coefficient commonly used in the field of similarity computation. As we know, the Jaccard coefficient measures similarity between sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. Here, the traditional definition of Jaccard is modified by replacing the intersection with the co-occurrence (i.e., $f(q,c)$) of the query q and the category c.

2.3.2.3. Overlap. The Overlap rank measure is defined as:

$$R(q,c) = \mathrm{Overlap}(q,c) = \frac{f(q,c)}{\min\{f(q), f(c)\}}. \quad (4)$$

The overlap coefficient is also a measure of agreement between two items or distributions. Different from Eq. (3), it uses the smaller page counts between $f(q)$ and $f(c)$ to normalize the occurrence $f(q,c)$.

2.3.2.4. Dice. The Dice rank measure is defined as:

$$R(q,c) = \frac{2 \times f(q,c)}{f(q) + f(c)}. \quad (5)$$

Table 1
Terminologies.

q	A search query keyword
S	A Web search engine
C	A flat or hierarchical topical taxonomy
c	The name of a category within the topical taxonomy C
$f(q)$	T1he page counts for the search query q returned by S
$f(c)$	The page counts for the category name c returned by S
$f(q,c)$	The page counts for the search query "$q + c$", i.e. the number of web pages containing both q and c
$R(q,c)$	The rank score for the category c given the query q

Table 2
Naïve ranking strategy.

Input: an input query q and the set of target categories C		
Output: a rank list of the target categories		
1.	Submit the concatenated queries $q + c_i$ (c_i in C) to a search engine one by one and obtain their own page counts as $f(q,c_i)$;	$O(\|C\|)$
2.	Compute $R(q,c_i)$ scores of the target categories based on Eq. (1) (c_i in C);	$O(\|C\|)$
3.	Quicksort $R(q,c_i)$ BY DESCEND.	$O(\|C\|\log\|C\|)$

The Dice coefficient is a similarity measure related to the Jaccard coefficient, but gives twice the weight to agreements (intersection).

2.3.2.5. Pointwise mutual information. The pointwise mutual information (PMI) is defined as:

$$R(q,c) = \log_2\left(\frac{\frac{f(q,c)}{M}}{\frac{f(q)}{M}\frac{f(c)}{M}}\right). \quad (6)$$

PMI (or specific mutual information) is a measure of association used in information theory and statistics. Traditionally, the PMI of a pair of two discrete random variables quantifies the discrepancy between the probability of their coincidence given their joint distribution *versus* the probability of their coincidence given only their individual distributions and assuming independence. Here, we modify the traditional PMI by replacing the probability with the page counts.

$f(q)$, $f(c)$, and $f(q,c)$ in Eqs. (2)–(6) are added by one respectively for guaranteeing the non-zero variable used in log function. The parameter M, a constant value, is greater than any page counts we get in computation. The similarity results are in general insensitive to the setting of the parameter M based on the theoretical analysis in Cilibrasi and Vitányi (2007). In our experiments, we set M to be the constant value of 9,000,000,000 which is larger than any page-counts in our experiments.

2.3.3. Examples

In this section, we show four examples to estimate semantic similarity using Eqs. (1) and (2) from two classes of ranking strategies. The reason that we select Eq. (2) is that NGD computation involves all variables and constant (i.e., $f(q)$, $f(c)$, $f(q,c)$, and M) appearing in other four rank measures.

Example 1. The query is "apple" and two candidate target categories are "computers" and "fruit". We get the page counts from a search engine and then we have

$f(apple, computers) = 4{,}280{,}000$
$f(apple, fruit) = 755{,}000$

If using Eq. (1), the above statistics show that the search query with keyword *apple* is more related to the topic "computer" than "fruit" because $f(apple, computers)$ is larger than $f(apple, fruit)$.

Example 2. The query is "apple" and two candidate target categories become "computers\hardware" and "computers\multimedia", respectively. After getting the page counts, we have

$f(apple, computers + hardware) = 16{,}000{,}000$
$f(apple, computers + multimedia) = 1{,}780{,}000$

Note that this example uses a hierarchical category taxonomy. Thus the category is now expressed by the sequence of labels along the path from the root to the corresponding node, such as "computers + hardware". Using Eq. (1), the above statistics tell that the search query with keyword "apple" has higher probability to be classified in the topic "computer\hardware" than "computer\multimedia".

Example 3. Let q denote "apple", c_1 denote "computers" and c_2 denote "fruit". We have

$f(apple) = 492{,}000{,}000$
$f(computers) = 391{,}000{,}000$
$R(apple, computers) = \frac{1}{\mathrm{NGD}(q,c_1)} = 0.6614$
$f(fruit) = 147{,}000{,}000$
$R(apple, fruit) = \frac{1}{\mathrm{NGD}(q,c_2)} = 0.6349$

Using Eq. (2), the above statistics show that using the current Web as the collection of pages, the query *apple* is more related to the topic "computer" than "fruit".

Example 4. Let q denote "apple", c_1 denote "computers\hardware" and c_2 denote "computers\multimedia". We have

$f(computers + hardware) = 14{,}400{,}000$
$R(apple, computers + hardware) = \frac{1}{\mathrm{NGD}(q,c_1)} = 1.88$
$f(computers + multimedia) = 1{,}910{,}000$
$R(apple, computers + mulimedia) = \frac{1}{\mathrm{NGD}(q,c_2)} = 0.37$

Using Eq. (2), the above statistics says that the query "apple" has higher probability to be classified in the topic "hardware" than "multimedia" under the same parent node "computer".

2.3.4. Discussions

Our approach uses page counts returned by a particular search engine to be an important components of semantic distance measures. As we know, search engines may be biased by their own search algorithms. Furthermore, Google acknowledges that the page counts given in a Google search are an estimate, but Google neither elaborates on the accuracy of this estimation nor reveals how it is calculated. Despite these issues, search engine still provides an excellent tool for research on different subjects (Cilibrasi & Vitányi, 2006; Cimiano & Staab, 2004; Gligorov, ten Kate, Aleksovski, & van Harmelen, 2007). For example, Cilibrasi and Vitányi (2006) uses page counts to train software by the semantic meaning of words, an important problem in Artificial Intelligence. Gligorov et al. (2007) used NGD as a weighting scheme for ontology matches. Different from these works, we work on the problem of search query classification and we empirically study the effect of a set of page-count based similarity measures on classification quality.

To obtain the true relative frequencies of words and phrase in society is a major problem in applied linguistic research. This requires analyzing representative random samples of sufficient sizes. The question of how to sample randomly and representatively is a continuous source of debate (Cilibrasi & Vitányi, 2007). We use the WWW and search engines in this paper. The same method may be used with other text corpora like the Oxford English Dictionary and frequency count extractors. In these cases one obtains a text corpus and frequency extractor biased semantics of the search terms. The Web is such a large and diverse text corpus, and search engine is such a great information extractor, we can view the relative page counts as an approximate of the true societal word and phrase usage. This point of view is receiving strong support in the state of art linguistic research (Keller & Lapata, 2003).

Table 3
Normalized ranking strategy.

Input: an input query q and the set of target categories C		
Output: a rank list of the target categories		
1.	Submit the input query q to a search engine and obtain its page counts as $f(q)$;	$O(1)$
2.	Submit the target categories (c_i in C) to a search engine one by one and obtain their own page counts as $f(c_i)$;	$O(\|C\|)$
3.	Submit the concatenated queries $f(q + c_i)$ (c_i in C) to a search engine one by one and obtain their own page counts as $f(q,c_i)$;	$O(\|C\|)$
4.	Compute $R(q,c_i)$ of the target categories based on one of Eqs. (2)–(6)(c_i in C);	$O(\|C\|)$
5.	Quicksort $R(q,c_i)$ BY DESCEND.	$O(\|C\|\log\|C\|)$

Another discussion is that we can expand the names of categories from other sources. For example, when submitting the name of a category into a dictionary software, e.g., WordNet, we can get related words which include the synonyms of the given category. The expansion method can provide relevant contexts of categories that can be easily added into our approach. In addition, if query log data is available, we can make use of the contexts of individual users to solve the problem of polysemy, such as mouse for animal or device, thus personalizing topical classification of queries. Since the goal of this paper is to study whether page counts are useful for query classification, expansion and personalization will be interesting topics in our future work.

2.4. Time complexity analysis

In this section, we analyze the time complexity of our approach. Our approach has no needs on training stage and manual classification data. The pseudo codes of the two classes of ranking strategies are given in Tables 2 and 3, respectively.

We first get page counts from a search engine (Step 1 in Table 2 and Steps 1–3 in Table 3). Then a rank measure produces the ranking scores of target categories given a query and a rank list of categories is generated based on the computed score (Step 2 in Table 2 and Step 4 in Table 3). Finally, the top N categories from the rank list will be returned to users as final classification results (the last step in both two tables).

As given in the two tables, the time complexity of the two kinds of ranking strategies are $O(|C|\log|C|)$ (C is the set of target categories) after we get word frequencies. The normalized ranking needs two more steps (Steps 1 and 2 in Table 3) to get $f(c)$ and $f(q)$ than the naïve ranking. The time complexity of both ranking strategies is depended on the number of target categories. Notice that the number of target categories also affect the time complexity of common classification approaches which usually compute the similarity between a query and each category and sort categories. When the number of categories becomes large, category selection (Shen et al., 2006) can narrow down the scope of the target categories. The benchmark of query classification provided by KDDCUP2005 used has 67 categories, a relative small number, so in the current stage, we do not need to take into account category selection.

In experiments, to get page frequencies, we have to download the result pages from a search engine. The runtime will be affected by the Internet traffic. If our approach is run in the server of a search engine, the time will be reduced a lot. In addition, our approach only needs the total number of related Web pages and there is no need to perform text processing like most conventional approaches (Pu et al., 2002; Shen et al., 2006; Shen et al., 2006).

Table 4
Examples of categories and queries.

Queries	
Applause	Msn homepage
Classroom accountability system	R.D. Call
Home theater equipment	Wedding food
Categories	
Computers\multimedia	Sports\baseball
Information\education	Entertainment\movies
Living\real estate	Entertainment\other

3. Experiments

In this section, we first introduce data sets and evaluation metrics. Then we present and discuss experiment results.

3.1. Data set

To test our approach, we conduct extensive experiments on the KDDCUP2005 data sets which are publicly available.[1] KDD Cup is the annual data mining and knowledge discovery competition organized by the ACM Conference on Knowledge and Data Discovery (KDD). Based on of the types of data collected, the task of KDDCUP varies from year to year. The competition in KDDCUP2005 is about classifying Internet user search queries and its task is to categorize 800,000 queries into 67 predefined categories. The meaning and intention of search queries are subjective. A search query "Python" might mean programming language to some people and snakes to others. Each participant was to classify all queries into as many as five categories. An evaluation set was created by human judges who independently evaluated and labeled 800 queries that were randomly selected from the sample of 800,000. We call this evaluation set as 800 data set. On average, the assessors assigned each query to 3.3 categories.

The target taxonomy is described in Fig. 2 which is hierarchical with two levels. Several examples of categories and queries are shown in Table 4. The queries may be only one word such as "applause" or a combination of words such "home theater equipment". The meanings of some words can be found in a traditional dictionary such as "wedding" and "food" while others have special meaning used on the Web such as "msn" representing Microsoft Network. The Web search queries are diverse in term length and topics. The target taxonomy is hierarchical with two levels. The top level has 7 categories. Each of the top level has several second level subcategories. The second level has 67 topics. They cover the areas in the Internet information space.

3.2. Evaluation measures and methodology

Precision, recall and F1 measures are the standard measures to evaluate the performance of classification in the literature. KDDCUP2005 give three classification evaluation metrics which are defined as:

$$\text{Precision} = \frac{\sum_i \#\text{ of queries correctly tagged as } c_i}{\sum_i \#\text{ of queries tagged as } c_i},$$

$$\text{Recall} = \frac{\sum_i \#\text{ of queries correctly tagged as } c_i}{\sum_i \#\text{ of queries labeled as } c_i \text{ by assessors}},$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

[1] http://www.sigkdd.org/kdd2005/kddcup.html

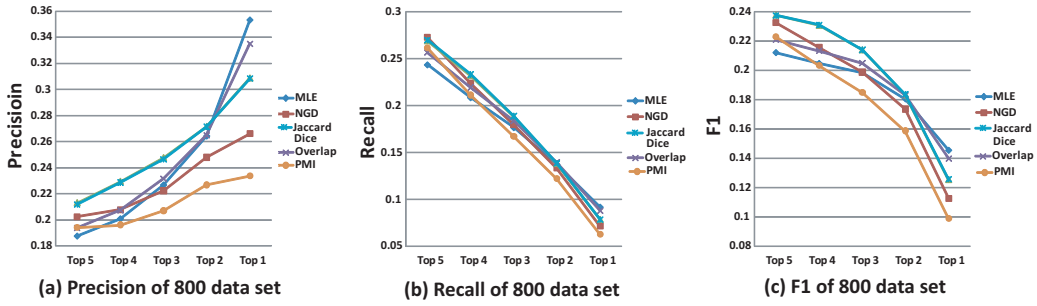

Fig. 1. Performances of different ranking measures varying with the number of classified categories.

The classification results produced by our approach are returned at the top N categories, e.g., at most five results required by KDDCUP2005 task. N as a parameter will be studied in our following evaluation.

As shown in Tabel 4, one issue we would like to point out about the KDDCUP 2005 category taxonomy is that each top level category has a special subcategory, called "other". This subcategory may be designed to include some unknown topics different from its siblings. The problem is that KDDCUP 2005 did not provide any description on each category and asked that the participants have to reply on common sense to acquire the implicit category semantics (Li, Zheng, & Dai, 2005). Therefore, we cannot know what set of topics were included in the subcategory "other".

Furthermore, from the user end, because uses usually do not have the whole taxonomy in their minds, even if the "entertainment\other" is shown in the user interface, users will not know the exact meaning of "other". The best they will do is to infer that the query posed by them belongs to "entertainment". Given that our classification approach needs to name a category in such a way that clearly represents the category's topic in order to get the corresponding page counts from a search engine. Therefore, the most reliable information we can get is that the subcategory "other" definitely belongs to its parent. Our evaluation in Sections 3.3.1 and 3.3.2 will first give the experiment results on the 60 subcategories without considering "other". However, the judgments provided by KDDCUP2005 include the subcategory "other" and other subcategories with specific topic names. For comparison with some popular solutions in the literature, in Section 3.3.3 we will estimate the page counts for the subcategory "other" using an advanced search operator and then do evaluation again.

3.3. Experimental results

In this section, we report four sets of experiments. First, we investigate how the setting of the parameter N will affect classification performance. Second, we use the naïve ranking strategy as the baseline and compare it with the normalized ranking strategy. To the best of our knowledge, no researchers study how to adaptively classify search queries to the changes of target categories except Shen et al. (2006). Their approach, however, still heavily relies on a fixed intermediate taxonomy, which is not really flexible. Since no suitable baselines can be compared with our approach in the situation, we then show the performance differences between our classification approach and some state-of-the-art solutions from KDD cup 2005 and a late publication. In addition, we also extract extra information from a search engine to see how much our approach can be further improved. Finally, we analysis the runtime of our approach.

3.3.1. Effect of parameter N

Our experiments use page counts from a popular search engine which is originally a subject directory of sites and it now is a search engine. The large directory databases hold by it is one of its strength as a search engine. We also got page counts from other two commercial search engines. The two search engines show similar performance but worse results than the search engine used by our experiments.

The parameter we study is the number of labels (i.e., N) assigned to each query. Fig. 1 shows the change trend of performance of different rank measures by varying the number of returned categories. For all six rank measures introduced in Section 2, the change trends of their performance are similar. When N (the number of returned categories) decreases from five to one, their precision scores (in the leftmost) increase. In contrast, their recall scores (in the middle) decrease significantly and their F1 values (in rightmost) decrease when N decreases from 5 to 1. For example, using Jaccard rank measure, the precision, recall and F1 scores are 0.2132, 0.2688, and 0.2378, respectively when $N = 5$. When N is changed to 1, the three scores become 0.3087, 0.0787, and 0.1254, respectively. In a word, we should assign few but accurate labels to achieve higher precision. If we want to get higher recall, we need to assign more possible labels.

3.3.2. Comparisons of ranking strategies

From Fig. 1, we find that the six rank measures differ a lot in terms of precision, especially when N varies from 5 to 1, the gap becomes larger. While they show similar results in terms of recall. The Jaccard and Dice rank measures generally achieve the best performance among all the six rank measures and their result lines are overlapped together. Except at Top 5, PMI rank measure produces the lowest F1 scores, even worse than MLE rank measure without normalization. Furthermore, MLE, the naïve ranking shows unstable results. At top 1, MLE generates the highest precision score. While at top 5, the precision scores of MLE decrease largely. Therefore, from the results in Fig. 1, we think that it is necessary to study different rank measures in order to select the best one for different system requirements on N.

From a quantitative view, we report the performance of different ranking measures at Top 5 in Table 5 because each query is labelled by at most five categories. We can see that Jaccard shows best results. In terms of precision, Jaccard achieves 13.5% improvement over MLE. In terms of F1, Jaccard achieves 12.1% improvement over MLE. Dice produces comparative performance with Jaccard. Overall, the normalized ranking strategy is more effective than the naïve ranking strategy. In addition, studying different normalization methods are needed for various applications.

3.3.3. Performance differences with other solutions

We show the performance differences with Beitzel's approach (Beitzel et al., 2007) and the corresponding performance scores from KDDCUP2005 report (Li et al., 2005) which are popular state-of-the-art solutions. Different from above experiments, we have to take into account the subcategory "other" in our evaluation. "other" definitely belongs to its parent and includes some unknown

Table 5
Performance of ranking measures at Top 5. The best two measures are shown in boldface.

	MLE	NGD	Jaccard	Overlap	Dice	PMI
P	0.1879	0.2025	**0.2132**	0.1941	**0.2121**	0.1943
R	0.2435	0.2725	0.2688	0.2563	0.2698	0.2614
F1	0.2121	0.2323	**0.2378**	0.2209	**0.2375**	0.2229

Table 6
Performance of different ranking measures at Top 5 with "Other". The best two measures are shown in boldface.

	MLE	NGD	Jaccard	Overlap	Dice	PMI
P	0.1728	0.198	**0.2074**	0.1770	**0.2064**	0.1905
R	0.2202	0.2576	0.2576	0.2300	0.2589	0.2479
F1	0.1936	0.2239	**0.2298**	0.2000	**0.2297**	0.2154

topics different from its siblings. For example, "computers\other" means that the queries in this subcategory are classified into "computers", but not "computers\software", "computers\hardware", and other siblings. Since we do not know what topics are included in "computers\other", we cannot directly use the topics as keywords to get their page counts. We estimate the page counts by using "computers" as the search keyword and adding a - (minus sign) to the beginning of each of its siblings. The advanced search operator "−" can exclude Web pages that have the names of its siblings. However, such estimating method puts strict constraint on the Web pages that can be returned. The Web pages in "computers \other" may includes the names of its siblings. The approximate page counts of "computers \other" will hurt our classification performance. However, to compare with popular solutions in the literature, we still include the evaluation on the subcategory "other".

We first report the experiment results of our ranking strategies with considering the subcategory "other" in Table 6. We find that Jaccard rank measure still shows highest performance. In terms of precision, Jaccard achieves 20.0% improvement over MLE. In terms of F1, the improvement of Jaccard is 18.7% over MLE. While Dice still produces comparative performance with Jaccard. Although the scores of precision, recall, and F1 in Table 6 are lower than those in Table 5, the improvement percentages reported in Table 6 are much more significant than those got from Table 5. We can know that the normalized ranking strategy is more robust than the naïve ranking strategy.

Then using Jaccard rank measure we show the performance differences with a Selectional Preference based approach (SP) proposed by Beitzel et al. (2007) and solutions in KDDCUP2005. The performance scores are given in Table 7 where the values of the column "KDD mean" are averaged by all the 37 submitted solutions in KDDCUP2005. As seen in Table 7, our approach (Jaccard) is able to produce close performance to SP and the average performance of the KDDCUP2005 participants for both precision and F1. Moreover, our recall is higher than the KDDCUP2005 average (i.e., 0.2576 vs. 0.2188). The reason we use the average value is that our goal is different and much more difficult than those popular solutions. The approach proposed by us can adaptively classify search queries into different types of target categories, which prohibits us from using external resources, such as WordNet, retrieved Web pages, search engine directories to enrich the features of queries and train a classifier beforehand. The participants in KDDCUP2005 competition made frequent use of these expensive outside resources for achieving high performance.

Table 7
Performance differences with other solutions at Top 5. The highest performance values are shown in boldface.

	SP	KDD mean	Jaccard	Jaccard + extra
Precision	0.2226	0.2545	0.2074	**0.2523**
Recall	0.2626	0.2188	0.2576	**0.3282**
F1	0.2409	0.2353	0.2298	**0.2853**

Moreover, the selectional preference based solution in Beitzel et al. (2007) needs a large amount of manually tagged queries as the training set, which is very expensive and involves heavy human labor, especially when a larger amount of labeled queries is needed to cope with the changes of target categories which may be caused by highly dynamics of the Web content and Web users' search behavior. Therefore, it is encouraging to see our flexible approach performed as well as it did and is easily adaptive to the change of target categories.

We also report the experiment results produced by adding retrieved Web pages as extra feature information into our approach, i.e., Jaccard + Extra in the rightmost column of Table 7. Given the 800 testing queries and 67 categories, we download the snippets and titles of their top 100 search results as the features to represent them and the computation of similarity between queries and categories uses the cosine similarity measure. The cosine scores using snippets and titles produce a rank list of category which is combined with the rank list of our approach using Jaccard by the popular rank aggregation method, i.e., Borda count (Borda, 1781). The combined approach is still flexible to new categories we obtain the content of categories on demand. In addition, our original approach (Jaccard) has been improved by 24.15% in terms of F1. This indicates that our original approach using page counts works in a different way as the method using retrieved Web pages, and they are complimentary. Notice that the combined approach outperforms SP by 18.4% and the KDDCUP2005 averaged performance by 21.2% in terms of F1. SP not only needs to manually label queries beforehand, but also use a large web search engine query log as a source of unlabeled data to aid in automatic query classification. In addition, SP and the approaches proposed in KDDCUP2005 did not discuss the flexibility problem.

3.3.4. Runtime analysis

In this section, we analyze the runtime of our approach. As shown in Tables 2 and 3, our approach need two main stages: (1) get page counts of a search engine and (2) compute ranking scores of target categories and sort them. A report [2] from Google Official Blog says that ranking is done by Google in a few milliseconds. Therefore, if our approach is run on the server of a search engine, the time of first stage can be very short since it will not be affected by unwanted Internet traffic and searching like client side.

After getting the page counts, we compute ranking scores for target categories given a query. Averaged by the six ranking measures, the classification time for each query is about 0.1 millseconds on a PC with 2G memeory and a 3.0 GHz CPU. If we use our own PC as a client to download the result pages. For each query, the total time of getting $f(q)$, $f(c)$, and $f(q,c)$ (the number of c is 67) is about 34s during our experiment time. The runtime of the first stage relies on the respond speed of the search engine and Internet speed (it can be reduced a lot when running on the server of a search engine). To best of our knowledge, no previous works on query classification report the time cost of training a classifier including data collecting and model building. If taking into considering the time cost of training, our approach only needs to retrieve the result page to get page counts while popular solutions like those competing in KDDCUP2005 not only require the result pages, but also take time to crawl the contents of returned Web pages from search engines and online taxonomies like ODP. As a result, our approach is fast when considering all the time cost from start to end.

[2] http://googleblog.blogspot.com/2008/05/introduction-to-google-search-quality.html

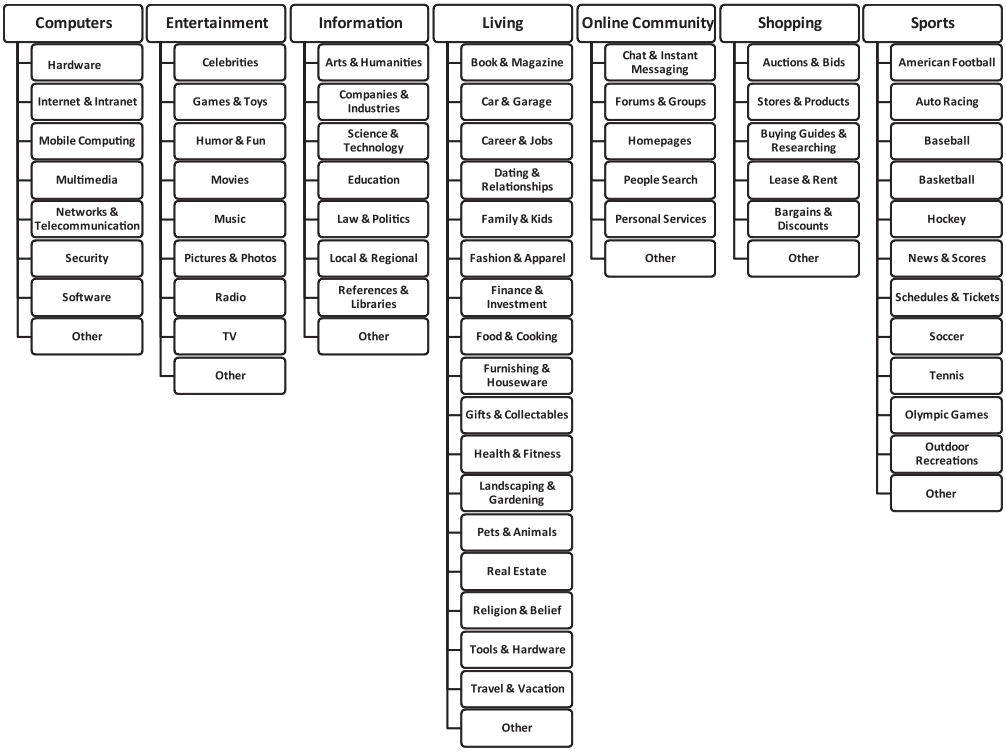

Fig. 2. Target taxonomy in KDDCUP2005 task.

4. Related work

There are different kinds of query classification problems in the literature. We discuss them from the following three aspects.

4.1. General classification of search queries

Gravano, Hatzivassiloglou, and Lichtenstein (2003) used a variety of state-the-art machine learning techniques to automatically classify search queries according to their locality. They discussed how their query categorization can help improve query result quality. For example, if underspecified queries such as "houses for sale" can be detected, they can be subsequently modified by adding the most likely target geographical location or by getting further user input to customize the results.

Recent works (Backstrom, Kleinberg, Kumar, & Novak, 2008; Wang et al., 2005) further discuss the spatial variation in queries like dominant locations, broader regional or national appeals. Kowalczyk, Zukerman, and Niemann (2004) studied how query classification can improve the quality of the question–answering process. They classified queries into six broad categories which represent the type of the desired answer, such as location, name, number, person, time, and so forth. Their results show that retrieval performance can be improved by dynamically adjusting the retrieval process on the basis of automatically learned query classes.

A number of researchers (Jansen, Booth, & Spink, 2007; Lee et al., 2005; Rose & Levinson, 2004) have automatically classified queries into informational, navigational and transactional types (or very similar taxonomies) which was first proposed by Broder (2002). Kang and Kim (2003) and Fujii (2008) further studied different retrieval methods depending on the query type, achieving at best modes improvements in retrieval effectiveness.

4.2. Topical classification of search queries

Manual topical classification is expensive since it will involves human labor, especially when the meanings of queries are evolving over time. Until recently there are not much published research that specifically addressed the problem of automatic topical classification of search queries. Some works (Beitzel et al., 2007; Pu et al., 2002) first manually classified a number of seed queries extracted from some query logs, and then used these classified queries as bridge to assign categories for left unlabeled queries. For example, Pu et al. (2002) assigned categories for an unkonwn query based on the categories of its cooccurring seed queries; Beitzel et al. (2007) proposed a selectional preference based method which tries to exploit some association rules between the query terms to help with the query classification.

However, it is a very difficult and time consuming task to provide enough pre-classified queries, especially when the target taxonomy is complicated. The task in KDDCUP 2005 (Li et al., 2005) is to automatically classify 800,000 queries to 67 predetermined categories with only 111 manually labelled queries. Most solutions submitted by 32 teams (Kardkovács, Tikk, & Bánsághi, 2005; Shen et al., 2006; Vogel et al., 2005) gathered extra information to augment query terms. Although the participants were able to achieve encouraging results, with the median F_1 score at 0.23 (max. 0.44) and a median precision of 0.24 (max. 0.75), participants mostly utilized the unlimited knowledge base available on the World Wide Web and search engines, such as search result snippets, search engine directories, search result Web pages, WordNet, and so on.

Recently, Broder et al. (2007) also determined the topic of a query by classifying the Web search results retrieved by the query. Using those solutions to classify queries is computationally prohibitive for a Web search engine, where the query volume can reach hundreds of millions per day (Sullivan, 2006). Another problem is caused by the ongoing changes in Web, target taxonomy. They

have to retrain a new classifier for these changes. To solve this problem, Shen et al. (2006) built a bridging classifier on the intermediate taxonomy ODP which is fixed, thus it is not really flexible.

Our novel feature-free approach can solve the above three problems. First, we do not require any training data. Second, we only extract the Web page counts from a search engine without crawling and processing the contents of search results. Last, the process of retrieving page counts can be done online, which are the most latest information and adaptive to the changes of target categories.

4.3. Query clustering

An alternative to classifying queries into categories is to allow class groups to emerge from a query set itself through clustering or other unsupervised learning methods. In query clustering, the problem is still lack of features in an individual query. Glance (2001) introduced a software agent that collected queries from previous users, and determined the query-to-query similarity based on the overlap of the URLs of Web pages returned by queries. Recent studies (Baeza-Yates, Hurtado, & Mendoza, 2007; Beeferman & Berger, 2000; Shi & Yang, 2007; Wen, Nie, & Zhang, 2002; Zhang & Nasraoui, 2006) used additional sources to enrich short search queries. There are two kinds of feature spaces commonly used in the literature: (1) content-sensitive (Baeza-Yates et al., 2007; Wen et al., 2002) and (2) content-ignorant features (Beeferman & Berger, 2000; Glance, 2001). Baeza-Yates et al. (2007) find related queries based on the content of clicked Web pages using click frequency as a weighting scheme. Their experiments show that using the content information of a Web page (e.g., nouns) is a more accurate query enrichment way to measure query similarity than using the URL of a Web page. Some studies (Shi & Yang, 2007; Zhang & Nasraoui, 2006) went another direction and they viewed Web logs as a set of transactions where a single submits a sequence of related queries in a time interval.

5. Conclusions and future work

In this paper, we have presented a novel feature-free flexible approach for search query classification using semantic distance. By feature free, we mean that our approach only utilizes queries and target categories themselves, and their page counts to perform classification. The page count is a key factor in semantic distance measures. By flexible, we mean that our approach can flexibly adapt to different category structure changes. By fast, we mean that our approach is very efficient with time complexity of $O(|C|\log|C|)$. We conducted extensive experiments on the benchmark data set provided by KDDCUP2005 and studied six popular rank measures. The experiment results show that Jaccard as a normalized ranking strategy is more effective than MLE as a naïve ranking strategy. Our approach produces close precision and F1 scores to the state of art solutions in search query classification and higher recall scores than the average values of KDDCUP2005 solutions. Those popular solutions did not address the problem of flexibility. Furthermore, after adding extra features, our performance is further improved.

Our research continues along a number of dimensions. First, we are interested in investigating the effectiveness of our approach by utilized the combined Web search results from multiple search engines in terms of better coverage and higher robustness. In the future, we will study the effectiveness of our approach using a larger set of categories with more sophisticated hierarchical structure. We believe that when the set of target categories becomes very large, e.g., ODP with more than 590,000 categories, category selection is an effective preprocessing to narrow down the scope of candidate categories.

References

Backstrom, L., Kleinberg, J. M., Kumar, R., & Novak, J. (2008). Spatial variation in search engine queries. In *Proceedings of the 17th international conference on World Wide Web (WWW'08), Beijing, China* (pp. 357–366).

Baeza-Yates, R. A., Hurtado, C. A., & Mendoza, M. (2007). Improving search engines by query clustering. *JASIST, 58*(12), 1793–1804.

Beeferman, D., & Berger, A. L. (2000). Agglomerative clustering of a search engine query log. In *Proceedings of the 6th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'00), Boston, MA, USA* (pp. 407–416).

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D. A., & Frieder, O. (2004). Hourly analysis of a very large topically categorized web query log. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'04)*, Sheffield, UK (pp. 321–328).

Beitzel, S. M., Jensen, E. C., Lewis, D. D., Chowdhury, A., & Frieder, O. (2007). Automatic classification of web queries using very large unlabeled query logs, ACM Transactions on Information Systems, 25 (2).

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th international conference on World Wide Web (WWW'07)*, Banff, Alberta, Canada (pp. 757–766).

Borda, J. (1781). Mémoire sur les élections au scrutin. *Comptes rendus de l'Académie des sciences, 44*, 42–51.

Broder, A. Z. (2002). A taxonomy of web search. *SIGIR Forum, 36*(2), 3–10.

Broder, A. Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., & Zhang, T. (2007). Robust classification of rare queries using web knowledge. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'07), Amsterdam, The Netherlands* (pp. 231–238).

Chen, C.-M., Lee, H.-M., & Chang, Y.-J. (2009). Two novel feature selection approaches for web page classification. *Expert Systems with Applications, 36*(1), 260–272.

Cilibrasi, R., & Vitányi, P. M. B. (2006). Automatic meaning discovery using google. In *Kolmogorov Complexity and Applications, no. 06051 in Dagstuhl Seminar Proceedings, Schloss Dagstuhl, Germany*.

Cilibrasi, R., & Vitányi, P. M. B. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering, 19*(3), 370–383.

Cimiano, P., & Staab, S. (2004). Learning by googling. *SIGKDD Explorations, 6*(2), 24–33.

Fujii, A. (2008). Modeling anchor text and classifying queries to enhance web document retrieval. In *Proceedings of the 17th international conference on World Wide Web (WWW'08), Beijing, China* (pp. 337–346).

Glance, N. S. (2001). Community search assistant. In *Proceedings of the 2001 international conference on intelligent user interfaces (IUI'01), Santa Fe, NM, USA* (pp. 91–96).

Gligorov, R., ten Kate, W., Aleksovski, Z., & van Harmelen, F. (2007). Using google distance to weight approximate ontology matches. In *Proceedings of the 16th international conference on World Wide Web (WWW'07), Banff, Alberta, Canada* (pp. 767–776).

Gravano, L., Hatzivassiloglou, V., & Lichtenstein, R. (2003). Categorizing web queries according to geographical locality. In *Proceedings of the 2003 ACM CIKM international conference on information and knowledge management (CIKM'03), New Orleans, Louisiana, USA* (pp. 325–333).

He, X., Duan, L., Zhou, Y., & Dom, B. (2009). Threshold selection for web-page classification with highly skewed class distribution. In *Proceedings of the 18th international conference on World Wide Web (WWW'09)* (pp. 1081–1082).

Jansen, B. J., Booth, D. L., & Spink, A. (2007). Determining the user intent of web search engine queries. In *Proceedings of the 16th international conference on World Wide Web (WWW'07), Banff, Alberta, Canada* (pp. 1149–1150).

Jansen, B. J., Spink, A., Bateman, J., & Saracevic, T. (1998). Real life information retrieval: A study of user queries on the web. *SIGIR Forum, 32*(1), 5–17.

Kang, I.-H., & Kim, G.-C. (2003). Query type classification for web document retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'03), Toronto, Canada* (pp. 64–71).

Kardkovács, Z. T., Tikk, D., & Bánsághi, Z. (2005). The ferrety algorithm for the kdd cup 2005 problem. *SIGKDD Explorations, 7*(2), 111–116.

Keller, F., & Lapata, M. (2003). Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics, 29*(3), 459–484.

Kim, S.-M., Pantel, P., Duan, L., & Gaffney, S. (2009). Improving web page classification by label-propagation over click graphs. In *Proceedings of the 18th ACM conference on information and knowledge management(CIKM'09)* (pp. 1077–1086).

Kowalczyk, P., Zukerman, I., & Niemann, M. (2004). Analyzing the effect of query class on document retrieval performance. In *Advances in artificial intelligence, 17th Australian joint conference on artificial intelligence (AI'04), Cairns, Australia* (pp. 550–561).

Lee, U., Liu, Z., & Cho J. (2005). Automatic identification of user goals in web search. In *Proceedings of the 14th international conference on World Wide Web (WWW'05), Chiba, Japan* (pp. 391–400).

Li, Y., Zheng, Z., & Dai, H. K. (2005). Kdd cup-2005 report: facing a great challenge. *SIGKDD Explorations, 7*(2), 91–99.

Özel, S. A. (2011). A web page classification system based on a genetic algorithm using tagged-terms as features. *Expert Systems with Applications, 38*(4), 3407–3415.

Pu, H.-T., Chuang, S.-L., & Yang, C. (2002). Subject categorization of query terms for exploring web users' search interests. *JASIST, 53*(8), 617–630.

Rose, D. E., & Levinson, D. (2004). Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web (WWW'04), New York, NY, USA* (pp. 13–19).

Shen, D., Yang, Q., Sun, J.-T., & Chen, Z. (2006). Building bridges for web query classification. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'06)*, Seattle, Washington, USA (pp. 131–138).

Shen, D., Pan, R., Sun, J.-T., Pan, J. J., Wu, K., Yin, J., et al. (2006). Query enrichment for web-query classification. *ACM Transactions on Information Systems, 24*(3), 320–352.

Shi, X., & Yang, C. C. (2007). Mining related queries from web search engine query logs using an improved association rule mining model. *JASIST, 58*(12), 1871–1883.

Sullivan, D. (2006). Searches per day (http://searchenginewatch.com/showPage.html?page=2156461 2006). URL http://searchenginewatch.com/showPage.html?page=2156461.

Vogel, D. S., Bickel, S., Haider, P., Schimpfky, R., Siemen, P., Bridges, S., et al. (2005). Classifying search engine queries using the web as background knowledge. *SIGKDD Explorations, 7*(2), 117–122.

Wang, L., Wang, C., Xie, X., Forman, J., Lu, Y., Ma, W.-Y., & Li, Y. (2005). Detecting dominant locations from search queries. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'05), Salvador, Brazil* (pp. 424–431).

Wen, J.-R., Nie, J.-Y., & Zhang, H. (2002). Query clustering using user logs. *ACM Transactions on Information Systems, 20*(1), 59–81.

Zhang, Z., & Nasraoui, O. (2006). Mining search engine query logs for query recommendation. In *Proceedings of the 15th international conference on World Wide Web (WWW'06), Edinburgh, Scotland, UK* (pp. 1039–1040).