Time-Dependent Semantic Similarity Measure of Queries Using Historical Click-Through Data[*]

Qiankun Zhao[†] Steven C. H. Hoi[‡] Tie-Yan Liu[§]
Sourav S. Bhowmick[†] Michael R. Lyu[‡] Wei-Ying Ma[§]

[†]School of Computer Engineering
Nanyang Technological University
50 Nanyang Ave, Singapore
qkzhao@pmail.ntu.edu.sg
assourav@ntu.edu.sg

[‡]CSE Department
The Chinese University of HK
Hong Kong, China
chhoi@cse.cuhk.edu.hk
lyu@cse.cuhk.edu.hk

[§]Web Search and Mining
Microsoft Research Asia
Beijing, China
tyliu@microsoft.com
wyma@microsoft.com

ABSTRACT

It has become a promising direction to measure similarity of Web search queries by mining the increasing amount of click-through data logged by Web search engines, which record the interactions between users and the search engines. Most existing approaches employ the click-through data for similarity measure of queries with little consideration of the temporal factor, while the click-through data is often dynamic and contains rich temporal information. In this paper we present a new framework of *time-dependent query semantic similarity model* on exploiting the temporal characteristics of *historical click-through data*. The intuition is that more accurate semantic similarity values between queries can be obtained by taking into account the timestamps of the log data. With a set of user-defined *calendar schema* and *calendar patterns*, our time-dependent query similarity model is constructed using the *marginalized kernel* technique, which can exploit both explicit similarity and implicit semantics from the click-through data effectively. Experimental results on a large set of click-through data acquired from a commercial search engine show that our time-dependent query similarity model is more accurate than the existing approaches. Moreover, we observe that our time-dependent query similarity model can, to some extent, reflect real-world semantics such as real-world events that are happening over time.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining*; H.3.3 [**Information System**]: Information Storage and Retrieval—*clustering, information filtering*

General Terms

Measurement, Performance, Experimentation

Keywords

click-through data, semantic similarity measure, marginalized kernel, event detection, evolution pattern

[*]This work was done when Qiankun Zhao and Steven C. H. Hoi were interns at Microsoft Research Asia.

1. INTRODUCTION

A real dilemma for most of the existing Web search engines is that users request for accurate search results while they only provide queries of limited length, which is usually less than two on average according to [19]. Recently, a lot of work has been done in the Web search community to expand the query terms with similar keywords for refining the search results [3, 6, 19, 20, 21]. The basic idea is to use the click-through data, which record the interactions between users and a search engine, as the feedback to learn the similarity between the query keywords. The existing work can be generally classified into two categories: *document term based query expansion* and *query term based query expansion*. The document term based query expansion, which was firstly introduced in [6] by Cui *et al.*, is to measure similarity between search queries and document terms based on the query log data of search engines. The basic idea is that if a set of documents is often selected for the same queries, then the terms in these documents are strongly related to the queries. Thus the probabilistic similarity between query terms and document terms can be calculated based on the query log data. The query term based query expansion [3, 21] is to measure similarity between query terms using the similarity propagation of Web pages being clicked. The intuition behind is that Web pages are similar if they are visited by users issuing similar queries, and queries are similar if the corresponding users visit similar Web pages.

1.1 Motivation

Although click-through data has received considerable attention on measuring similarity between query terms in the Web research community, most of existing work ignored an important fact, i.e., the similarity between query terms often evolves over time. Here the *similarities* between query terms are usually obtained by the similarity propagation between queries, pages, and their co-occurrences [21]. The dynamic nature of query terms similarity over time can be attributed to many factors, such as seasons, holidays, and special events, etc. This dynamic nature of query terms similarity is usually embedded in the click-through data implicitly. Traditional methods without temporal consideration have limitations to measure such underlying semantic similarity of query terms accurately.

Therefore, it becomes a challenging and important task to develop an effective model for similarity measure from the

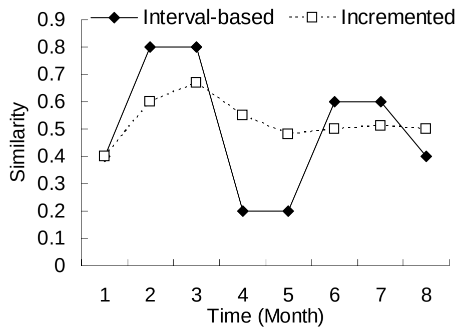

Figure 1: Incremented and interval-based methods for similarity measure

click-through data that can take advantage of the dynamic nature of queries and pages over time. In particular, two challenging issues are needed to be solved as follows:

- How to exploit the click-through data for semantic similarity measure of queries in terms of temporal consideration?
- How to develop an effective model that can reflect both explicit content similarity and implicit semantics between queries?

To address the challenging issues, let us first illustrate an example to compare two different approaches of measuring similarity between two queries over time in Figure 1. In the figure, the dotted line represents the first method that measures the similarity value at each time point based on all the click-through data available at that time. We refer to this method as the *incremented* approach. Different from the *incremented* approach, the other approach, as shown in the solid line, measures the similarity only by the click-through data given at that time interval. We refer to this method as the *interval-based* approach.

From the comparison, we see that the *interval-based* approach can better reflect the temporal factor than the *incremented* approach. For instance, in the *interval-based* approach, the similarity values of the second and third month are as high as *0.8*, while the similarity values in the fourth and fifth month are as low as *0.2*. But in the *incremented* approach, the similarity values from the second month to the fifth month are respectively *0.6, 0.68, 0.55,* and *0.48*. This shows that the similarity values in the *incremented* approach are relatively fixed, which usually cannot evidently reflect the dynamic nature of similarities between the query terms.

Hence, it is important to develop a similarity model that exploits the click-through data effectively in a *time-dependent* way. To this end, we propose a novel time-dependent framework to exploit the click-through data for semantic similarity measure between queries. An overview of our proposed framework is given in the following.

1.2 Overview

In this paper we suggest a time-dependent framework to measure the semantic similarity between Web search queries from the click-through data. More specifically, we propose a novel *time-dependent query term semantic similarity model*, which can exploit the click-through data more effectively than traditional approaches such as the incremented approach shown in Figure 1. Our time-dependent model monitors the terms' similarity over the temporal dimension and attempts to discover the *evolution pattern* with the click-through data. Note that in this paper, the term *evolution pattern* of the query terms' similarity refers to how the similarity values vary over time.

The basic idea of our solution is to construct a model from the click-through data by partitioning the click-through data into sequences of sub-groups with respect to certain predefined *calendar schema* and *calendar patterns.* For example, to monitor query similarity that may change on a daily basis, the click-through data is partitioned into sequences of subgroups, where each sequence consists of click-through data of an individual day. Then, using proposed semantic similarity measure methodology based on the *marginalized kernel function*, a daily based temporal similarity model can be constructed. As a result, the time-dependent model can accurately reflect the daily based patterns such as large or small values of query similarities during the specific days.

Our proposed model can be used for more accurate and efficient query expansion for Web search engines. Our empirical results with the real-world click-through data collected from a commercial search engine show that our proposed model can model the evolution of query terms similarity accurately. In summary, the contributions of our work in this paper can be summarized as follows:

- To the best of our knowledge, we proposed the first time-dependent model to calculate the query terms similarity by exploiting the dynamic nature of click-through data.
- A probabilistic framework for constructing the time-dependent query term similarity model is proposed with the *marginalized kernel*, which measures both explicit content similarity and implicit semantics from the click-through data.
- Extensive experiments have been done to evaluate the proposed similarity model using a large collection of click-through data collected from a commercial search engine.

The rest of the paper is organized as follows. In Section 2, we review related work of calculating the query terms similarity from the click-through data and temporal analysis of the click-through data. Section 3 presents our time-dependent framework for semantic similarity measure of query terms in the context of query expansion, in which the time-dependent model is formulated by a marginalized kernel. Section 4 provides our empirical evaluation, in which some empirical examples are shown from real-world click-through data and extensive experiments are conducted to evaluate the performance of our model. Section 5 discusses limitations and future work. Section 6 concludes this work.

2. RELATED WORK

In this section, we review related work mainly in the following two aspects: query expansion with click-through data and temporal analysis of click-through data.

Query expansion with click-through data is motivated by the well-known relevance feedback techniques, which modify the queries based on users' relevance judgments of the retrieved documents [1, 7, 10]. Typically, expansion terms are extracted based on the frequencies or co-occurrences of the terms from the relevant documents. However, it is difficult to obtain sufficient feedbacks since users are usually reluctant to provide such feedback information. Even though the pseudo-relevance feedback approach can partially alleviate the lack of feedbacks, it still suffers from the failure of the assumption that a frequent term from the top-ranked relevant documents will tend to co-occur with all query terms, which may not always hold [2, 20].

The click-through data has been studied for query expansion in the past [3, 6, 19, 21]. The existing work can be categorized into two groups. The first group is to expand queries with similar queries based on the assumption that similarity between queries may be deduced from the common documents the users visited by issuing those queries [3, 19, 21]. The second group is to expand queries with similar terms in the corresponding documents being visited in the history [6]. In addition to query expansion, click-through data has been used to learn the rank function as well [9, 12].

More recently, several efforts began to analyze the temporal and dynamic nature of the click-through data [4, 5, 13, 16]. In [4], Beitzel *et al.* proposed the first approach to show the changes of popularities on an hourly basis. With the categorization information of the Web queries, the results show that query traffic from particular topical categories differs from both the query stream as a whole and queries in other categories. Moreover, Shen *et al.* [13] proposed to investigate the transitions among the topics of pages visited by a sample of Web search users. They constructed a model to predict the transitions in the topics for individual users and groups of users. Vlachos *et al.* [16] suggested to identify similar queries based on the historical demand patterns, which are represented as time series using the best Fourier coefficients and the energy of the omitted components. Similarly, Chien and Immorlica [5] proposed to find semantically similar queries using the temporal correlation.

However, compared with the existing approaches above, our approach differs from them in the following important aspects. Instead of only discovering Web queries that have specific characteristics, such as burst and periodic queries in [16], our scheme models the similarity of queries based on the predefined *calendar schema* and *calendar patterns*. This makes our scheme more general in modelling the temporal feature of queries, which is difficult to be achieved by existing techniques. Further, our scheme also critically differs from the approach in [5], which attempts to figure out a unified similarity or correlation between two terms that may not exist for every case and may change rapidly in different situations of real-world environment. Moreover, different from other previous work on categorized Web queries [4, 13], which monitor the differences and transitions between queries from different categories in terms of the frequency of being issued, we focus on exploring queries in terms of their semantic similarity over time. Finally and importantly, in both [16, 5] and [4], only the frequency of the queries are considered, while in our approach both the queries and the corresponding pages being clicked are engaged, together with the relationships between queries and pages. Different from previous approaches, we propose a new time-dependent semantic similarity model formulated by the marginalized kernels in a probabilistic framework to explore explicit content similarity and implicit semantics from the click-through data very effectively.

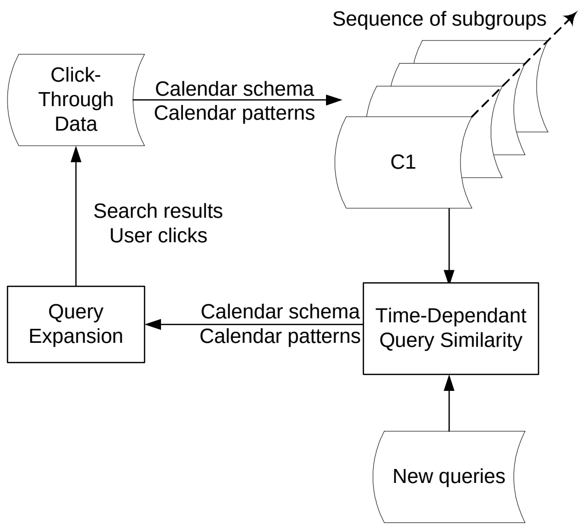

Figure 2: A Framework of Time-Dependent Semantic Similarity Measure Model between Queries

3. THE TIME-DEPENDENT FRAMEWORK

In this section, we propose a probabilistic framework to construct the time-dependent query similarity model. By exploiting the click-through data over time, a semantic similarity measure model is proposed based on the *marginalized kernel* technique [15]. Figure 2 illustrates the framework of our proposed semantic similarity model. The details of our framework will be discussed as the following. First, some preliminary definitions are given to describe click-through data and calendar pattern formally. Following these definitions, a probabilistic approach is proposed to measure content similarity between query terms. Then, we study an efficient clustering technique on the click-through data to acquire the implicit cluster semantics of the data. Lastly, the time-dependent semantic similarity model is formulated by the marginalized kernel function, which measures both explicit content similarity and implicit cluster semantics effectively from the click-through data.

3.1 Click-Through Data & Calendar Pattern

Click-through data, representing query log of Web search engines, keep the records of interactions between Web users and the searching engines. Similar to the transaction data in the supermarket, the click-through data consist of a sequence of users sessions in some format like <*IP address, query, clicked page, time*>. In the literature, there are two different ways to represent the click-through data. One way is to represent the click-through data as session databases where each session represents a pair of a query and a page that the user issued and clicked (hereafter, we call such pairs as *query-page pairs*) as shown in Table 1 [6, 19]. Note that the IP addresses are not shown in the table due to the privacy issue. Recently, some variants of this representation have been proposed by taking into account the rank of the

IP address	Query	Page	Time
xxx.xxx.xxx	BMW	http://www.bmw.com	14:02 02112005
xxx.xxx.xxx	SVM	http://svm.first.gmd.de	15:31 03252005
xxx.xxx.xxx	MSN	http://www.MSN.com	21:14 02142005

Table 1: Transaction representation of click-through data.

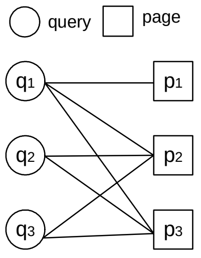

Figure 3: Bipartite graph representation of click-through data.

page and the ID of the corresponding users [9, 14]. The other representation method is to use a bipartite graph, like the example in Figure 3, where the queries and pages are represented as two sets of nodes and the query-page pair co-occurrence relationships are represented as the edges between the corresponding nodes [3, 21].

We use the session database approach to represent the click-through data. Each session is represented as a triple $< \vec{q}, p, \vec{t} >$, where $\vec{q}$ represents the query being issued, p represents the pages being clicked, and $\vec{t}$ represents the timestamp. Note that we use the vector representation for both the queries and timestamps. The reason is that, in this paper, the queries will be represented as a vector of related web pages that are led to by the queries. For the timestamps, with a fix *calendar schema*, they are represented as vectors as well. The key distinguishing feature of our query similarity measure is that rather than only using the query-page pairs, the corresponding timestamps are used as well. From the real query log data obtained from a commercial Web search engine, we observe that the granularity of the timestamps can be in mini-second. However, to analyze the temporal patterns of queries, the time granularity can be application dependent and query dependent. To make our framework flexible, we allow users to specify any types of *calendar-based patterns* they are interested in depending on their domain knowledge and application requirements. Formally, the *calendar schema* and *calendar-based pattern* are defined as follows:

Definition 1. **Calendar Schema:** *A calendar schema, $S = (R, C)$, is a relational schema R with a constraint C, where $R = (f_n : D_n, f_{n-1} : D_{n-1}, \cdots, f_1 : D_1)$, C is a Boolean valid constraint on $D_n \times D_{n-1} \times \cdots \times D_1$ that specifies which combinations of the values in $D_n \times D_{n-1} \times \cdots \times D_1$ are valid.* □

Here each attribute f_i in the relation schema is a calendar unit name such as year, month, week, day, hour, etc. Each domain D_i is a finite subset of positive integers. C is a Boolean function specifying which combinations of the values in $D_n \times D_{n-1} \times \cdots \times D_1$ are valid. For example, we may have calendar schema *(year: {2000, 2001, 2002}, month: {1, 2, 3, $\cdots$,12}, day: {1, 2, 3, $\cdots$, 31})* with the constraint that evaluate $< y, m, d >$ to be "true" only if the combination gives a valid date. For instance, $<$ *2000, 2, 15* $>$ is valid while $<$ *2000, 2, 30* $>$ is invalid. The reason to use the calendar schema here is to exclude invalid time interval due to the combinations of calendar units. Moreover, by modifying the constraint, users can further narrow down valid time intervals for application and query dependent reasons. Hereafter, we use $*$ to represent any integer value that is valid based on the constraint. For instance, if we use $*$ to represent months in with the above mentioned calendar schema, then $*$ refers to any integer value from *1* to *12*.

Definition 2. **Calendar Pattern:** *Given a calendar schema $S = (R, C)$, a calendar pattern, denoted as CAP, is a tuple on R of the form $< d_n, d_{n-1}, \cdots, d_1 >$ where $d_i \in D_i \cup \{*\}$.* □

For example, assume we are given the calendar schema $< year, month, day >$, then the calendar pattern $< *, 1, 1 >$ refers to the time intervals "the first day of the first month of every year". Similarly, $< 2002, *, 1 >$ represents the time intervals "the first day of every month in year 2002."

3.2 Probabilistic Similarity Measure

Different from the previous query expansion approaches that use the query log and the actual Web pages to extract similarity between terms in the query space and terms in the document space [6, 13], we only employ the query log. We propose a dual approach of the existing information retrieval model by representing each query term as a vector of documents, namely $\vec{q} =< w_1, w_2, \cdots, w_n >$ in which w_i represents the projection weight on the i^{th} page. In our paper, this weight is calculated from the *Page Frequency (PF)* and *Inverted Query Frequency (IQF)*, which are formally defined as follows:

Definition 3. **PF.IQF:** *Given a query $\vec{q}$ and a Web page p_i that has been clicked by users who issued $\vec{q}$ via the search engine. Then, the Page Frequency (PF) and the Inverted Query Frequency (IQF) are defined as:*

$$PF(\vec{q}, p_i) = \frac{f(\vec{q}, p_i)}{\sum_j f(\vec{q}, p_j)}, \; IQF(p_i) = \log \frac{|\vec{q}|}{| < \vec{q}, p_i > |}$$

□

Here $f(\vec{q}, p_i)$ is the number of times that page p_i has been clicked by users who issued the query $\vec{q}$. $\sum_j f(\vec{q}, p_j)$ refers to the total number of times that the pages have been clicked by users who issued the query $\vec{q}$. $|\vec{q}|$ refers to the total number of times that the query $\vec{q}$ has been issued and $| < \vec{q}, p_i > |$ refers to the times that the page p_i has been clicked by users who issued $\vec{q}$. As a result, the weight of page p_i is calculated as $w_i = PF(\vec{q}, p_i) \times IQF(p_i)$.

Based on the document weight vector representation, the similarity between two queries in content can be defined by a cosine kernel function as follows.

Definition 4. **Content Similarity Measure:** *Given two queries $\vec{q}_1$ and $\vec{q}_2$, their probabilistic similarity in content, denoted as $K_{cos}(\vec{q}_1, \vec{q}_2)$, is defined as:*

$$K_{\cos}(\vec{q}_1, \vec{q}_2) = \frac{\vec{q}_1^T \vec{q}_2}{||\vec{q}_1|| \cdot ||\vec{q}_2||}$$

□

Note that in the above similarity measure, all occurrences of queries and Web pages are considered to be equally important and the timestamps are not used.

3.3 Time-Dependent Semantic Similarity Model

Based on the content similarity measure and the calendar patterns proposed in the previous section, we now present the framework of the time-dependent query semantic similarity model. Before going into the details of our framework, let us first define the relationship between the timestamp and the calendar pattern to facilitate our following discussions as follows:

DEFINITION 5. **Contained:** *Given a calendar pattern* $< d_n, d_{n-1}, \cdots, d_1 >$ *denoted as* CAP_i *with the corresponding calendar schema* $S = (R, C)$. *A timestamp* $\vec{t}$ *is represented as* $< d'_n, d'_{n-1}, \cdots, d'_1 >$ *according to* R. $\vec{t}$ *is* ***contained*** *in* CAP_i, *denoted as* $\vec{t} \prec CAP_i$, *if and only if* $\forall\ 1 \leq l \leq n$, $d'_l \in d_l$. □

For example, given a calendar pattern $< *, 2, 12 >$ with the calendar schema $< week, day\ of\ the\ week, hour >$, then the timestamp *2005-09-30 12:28* is not contained in this calendar pattern as it is not the second day of the week, while the timestamp *2005-09-26 12:08* is.

DEFINITION 6. **Click-Through Subgroup (CTS):** *Given a calendar schema* S *and a set of calendar patterns* $\{CAP_1, CAP_2, \cdots, CAP_m\}$, *the click-through data can be segmented into a sequence of click-through subgroups (CTSs)* $< CTS_1, CTS_2, \cdots, CTS_m >$, *where all query-page pairs* $< \vec{q}, p_i, \vec{t_i} > \in CTS_l$, $\vec{t_i} \prec CAP_l$, $1 \leq l \leq m$. □

With the above definitions, in general, given a collection of click-through data, we can first partition the data into sequences of CTSs based on the user-defined calendar pattern and the corresponding timestamps. For example, given a weekly based calendar schema $< week, day >$ and a list of calendar patterns $< *, 1 >, < *, 2 >, \cdots, < *, 7 >$, the click-through data will be partitioned into sequences of 7 CTSs $< CTS_1, CTS_2, \cdots, CTS_7 >$, where CTS_i represents the group of click-through data whose timestamps are contained in the i^{th} day of the week.

After that, the query similarities are computed within each subgroup and are aligned into a sequence to show the patterns of historical change. At the same time, a model is generated, with which we can obtain the query similarities by inputting queries and timestamps. Given the above example, we can obtain the query similarity on each day of the week. Moreover, we can monitor how the query similarity changes over time within each week in a daily basis. Also, given two queries and the day of a week, the query similarity can be returned. Then, the process iterates for presenting the results and collecting the click-through data with users interactions. Hereafter, we focus on how to construct the time-dependent query similarity model based on sequences of click-through subgroups.

In order to learn the implicit semantics embedded in the click-through data, we first apply clustering techniques on the data to find the cluster information in each click-through subgroup. When the cluster results are obtained, we then formulate our semantic similarity model by the marginalized kernel technique that can unify both the explicit content similarity and the implicit cluster semantics very effectively. Before the discussion of our semantic similarity model, we first discuss how to cluster the click-through data efficiently. Let us first give a preliminary definition.

DEFINITION 7. **Clusters of Click-Through Pages:** *Given a click-through subgroup, we can obtain clusters of click-through pages* $\Omega = \{c_1, c_2, \cdots, c_k\}$ *by grouping the pages that are similar in semantics, where* k *is determined by some clustering algorithm.* □

In the literature, some clustering methods have been proposed to cluster Web pages in the click-through data using the query-page relation and propagation of similarities between queries and pages [3, 21]. In [3], an agglomerative clustering method is proposed. The basic idea is to merge the most *similar* Web pages and queries iteratively. Originally, the *similarity* is defined based on the overlaps of neighbors in the bipartite graph representation of the click-through data as shown in Figure 3.

For the efficiency reason, we adopt the agglomerative clustering method in [3]. In our clustering approach, neighbors in the bipartite graph are assigned with different weights instead of being taken as equal. The intuition is that the strength of the correlation between two query-page pairs may be quite different. For example, the strength of a query-page pair that co-occurs once should not be as equal as a query-page pair that co-occurs thousands of times. Hence, we represent the weights of the neighbors based on the number of times the corresponding query-page pairs co-occur. That is, the weight of a page for a given query is the number of times that page was accessed against the total number of times the corresponding query was issued. Similarly, the weight of a query for a given page is the number of times the query was issued against the total number of times the corresponding page was visited. Then each query is represented as a vector of weighted pages, and each page is represented as a vector of weighted queries. As a result, similarities between pages or queries are calculated based on the cosine similarity measure.

More details about the clustering algorithm can be found in [3]. Note that the clustering algorithm is applied on each of the click-through subgroups. Based on the clustering results, we now introduce the marginalized kernel technique, which can effectively explore the hidden information for similarity measure in a probabilistic framework [11, 15].

DEFINITION 8. **Marginalized Kernel:** *Assume that a visible variable* x *is described as* $x \in X$, *where the domain* X *is a finite set. Suppose a hidden variable* h *is described as* $h \in H$, *where* H *is a finite set. A joint kernel* $K_Z(z, z')$ *is defined between the two combined variables* $z = (x, h)$ *and* $z' = (x', h')$. *The* ***marginalized kernel*** *in* X *is defined by taking the expectation with respect to the hidden variables as follows:*

$$K(x, x') = \sum_{h \in H} \sum_{h' \in H} p(h|x) p(h'|x') K_Z(z, z')$$

□

In the above definition, the terms $p(h|x)$ and $p(h'|x')$ are employed to describe the uncertainty of the hidden variables h and h' related to the visible variables x and x', respectively. The marginalized kernel models the probability of similarity between two objects by exploiting the information with the hidden representations. Given the above definition of the marginalized kernel function, we employ it to formulate our time-dependent kernel function for semantic similarity measure of queries as follows.

Definition 9. **Time-Dependent Query Semantic Similarity Measure:** *Given two queries $\vec{q}$ and $\vec{q}'$, together with a specific timestamp $\vec{t}$, the time-dependent semantic similarity between the two queries is measured by a time-dependent marginalized kernel function $K_T(\vec{q}, \vec{q}'|\vec{t})$ as follows:*

$$\begin{aligned} & K_T(\vec{q}, \vec{q}'|\vec{t}) \\ & = \sum_{\forall c}\sum_{\forall c'} K_Q\left(Q_{c|t}, Q'_{c'|t}\right) p(c|q,t)p(c'|q',t) \\ & = K_{\cos}(q, q'|t)\left(\sum_{\forall c}\sum_{\forall c'}\varphi(c,c'|t)p(c|q,t)p(c'|q',t)\right) \\ & = K_{\cos}(q, q'|t)\left(\sum_{c\in\Omega(t)} p(c|q,t)p(c|q',t)\right) \\ & = \frac{q_t \bullet q'_t}{||q_t|| \times ||q'_t||}\left(\sum_{c\in\Omega(t)} p(c|q,t)p(c|q',t)\right) \end{aligned}$$

where c and c' are the guessed clusters given the queries, $Q_{c|t} = (q, c|t)$ and $Q'_{c'|t} = (q', c'|t)$. K_Q is a joint kernel, $\varphi(c, c'|t)$ is a function whose value is equivalent to 1 if c and c' are the same and 0 otherwise, and q_t and q'_t are time-dependent query vectors. □

In the above formulation, the joint kernel $K_Q\left(Q_{c|t}, Q'_{c'|t}\right)$ is defined on the two combined query variables as follows:

$$K_Q\left(Q_{c|t}, Q'_{c'|t}\right) = \varphi(c, c'|t)K_{\cos}(q, q'|t),$$

where $\varphi(c, c'|t)$ is a function to indicate whether c and c' are the same cluster of click-through data, and $K_{\cos}(q, q'|t)$ is a time-dependent joint cosine kernel on the two time-dependent query vectors $K_{\cos}(q, q'|t) = \frac{q_t \bullet q'_t}{||q_t|| \times ||q'_t||}$. Note that the query vectors are only computed on the subgroup CTS_i, to which the given timestamp $\vec{t}$ belongs.

From the definition of time-dependent marginalized kernel, we can observe that the semantic similarity between two queries given the timestamp $\vec{t}$ is determined by two factors. One is the time-dependent content similarity measure between queries using the cosine kernel function; another is the likelihood for two queries to be grouped in a same cluster from the click-through data given the timestamp.

4. EMPIRICAL EVALUATION

In this section we conduct a set of empirical studies to extensively evaluate the performance of our time-dependent query semantic similarity model. In the rest of this section, we first describe the dataset used in our evaluation and the experimental setup in our experiments. Then, we show several empirical examples to illustrate the real-world results using our time-dependent framework. After that, we discuss the quality measure metric used in our performance evaluation. Finally, the quality of the time-dependent query similarity model is evaluated under different scenarios.

4.1 Dataset

A real click-through dataset collected from Microsoft MSN search engine is used in our experiments. The click-through data contains *15* million records of query-page pairs over *32* days from *June 16, 2005* to *July 17, 2005.* The size of the raw data is more than *22 GB.* Note that the timestamps for each transaction is converted to the local time using the information about the IP address. In the following experiments, the entire click-through data is partitioned into subgroups based on the user-defined calendar schema and calendar patterns. For instance, given the calendar schema $< hour, day, month >$ with the calendar pattern $< 1, *, * >$, $< 2, *, * >$, $\cdots$, $< 24, *, * >$, the click-through data is partitioned into a sequence of *24* subgroups, where each group consists of the query-page pairs occurred during a specific hour of everyday. Then, the average number of query-page pairs in each group is around *59,400,000.*

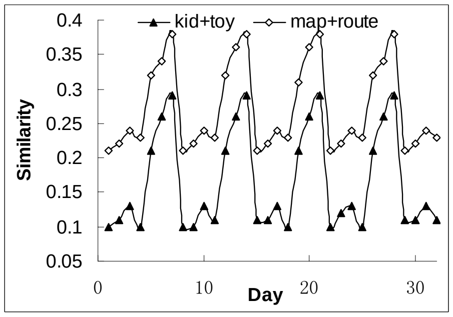

Figure 4: Daily-based query similarity evolution

4.2 Empirical Examples

In this subsection, we present a set of examples of query term similarity evolution over time extracted from the real click-through data collected from MSN search engine. As there are many different types of evolution patterns, here we present some of the representatives.

Figure 4 shows the similarities for two query pairs ("kid", "toy") and ("map", "route") on a daily basis in the *32* days. We observe that the similarities changed periodically in a weekly basis. That is, the similarities changed repeatedly: starting low in the first few days of the week and ending high in the weekend. To reflect such time-dependent pattern, we apply our time-dependent query similarity model to the two query pairs. Here the calendar schema and calendar patterns used are $< day, week >$ and $< 1, * >$, $< 2, * >$, $\cdots$, $< 7, * >$. Figure 5 shows the time-dependent query similarity measurement for the two query pairs in Figure 4. We can see that the time-dependent query similarity model can efficiently summarize the dynamics of the similarity over time on a weekly basis. However, as shown in Figure 6, the *incremented approach* cannot accurately reflect the highs and lows of the similarity values. Note that the calendar schema and calendar patterns used in the model are use-defined with related domain knowledge. With inappropriate calendar schema and calendar patterns, we may not be able to construct accurate time-dependent query similarity models. For instance, for the same query pairs, if we use $< hour, day >$ and $< 1, * >$, $< 2, * >$, $\cdots$, $< 24, * >$ as calendar schema and calendar patterns (shown in Figure 7). We can see that there are no predictable change patterns, hence there is no *hour of the day* based time-dependent model that can accurately model the similarity.

Figure 8 shows the similarity measurement for two query pairs ("weather", "forecast") and ("fox", "news") over one and a half day on hourly basis. We can see that box query pairs have two peak values in every day and this pattern repeatedly occur in the dataset. Based on this observation, we propose to model their similarity using the time-dependent query similarity model with a hourly based cal-

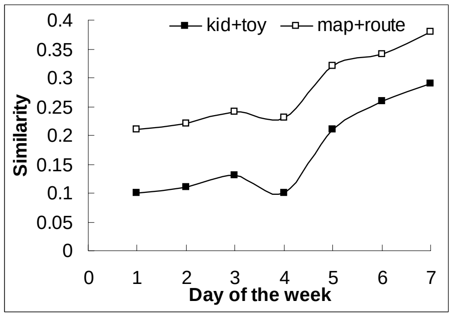

Figure 5: Weekly-based time dependent query similarity model

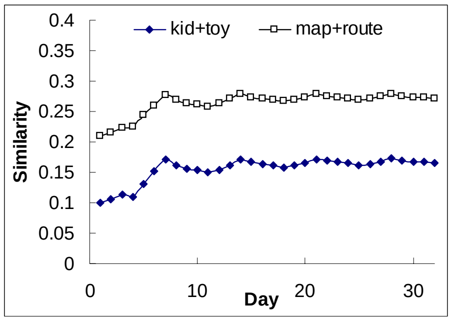

Figure 6: Query similarity with incremented approach

endar patterns. That is, the calendar schema and calendar patterns used are $< hour, day >$ with $< 1, * >$, $< 2, * >$, $\cdots$, $< 24, * >$. Figure 9 shows the time-dependent query similarity model. Similarly, Figure 10 shows the similarity values calculated using the *incremented approach*, which is clearly not accurate compare to the time-dependent similarity model.

Figure 11 shows the similarity measurement of two query pairs ("father", "gift") and ("firework", "show") on a daily basis. The corresponding similarity values extracted using the *incremented approach* are shown in Figure 12. We can see that the time-dependent model cannot be constructed for the two sets of query pairs from the data available in our collection. The reason is that to track event-based query pairs' similarity, e.g., the "father's day" based query pairs' similarity, we need at least years' of click-through data since such events happen only once every year. Note that the previous examples, which can be modeled using the time-dependent model, are within a time interval of *32* days such as weekly based and hour of the day based (our click-through dataset only contains data for *32* days).

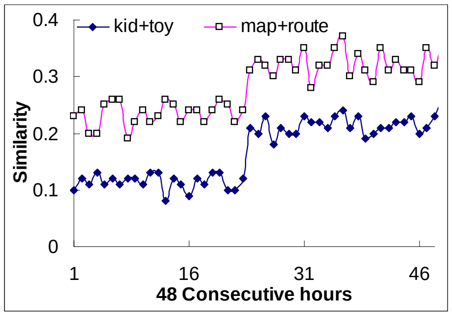

Figure 7: Hourly-based query similarity evolution

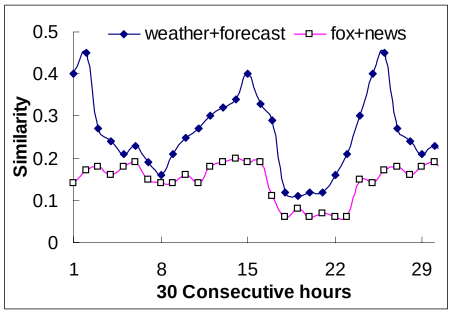

Figure 8: Hourly-based query similarity evolution

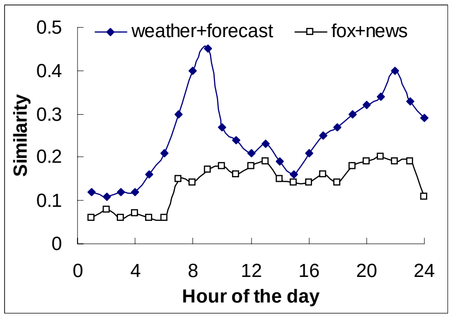

Figure 9: Hourly-based time dependent query similarity model

4.3 Quality Measure

To evaluate the quality of the time-dependent query similarity model, the dataset is partitioned into two parts. The first part consists of a collection of click-through data in the first few days, while the second part consists of the click-through data in the rest of *32* days. Note that the timestamps of click-through data in the first part must be earlier than the timestamps of the click-through data in the sec-

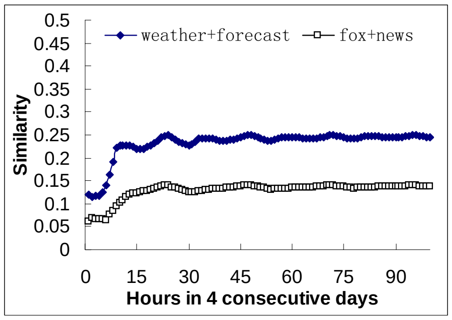

Figure 10: Query similarity with incremented approach

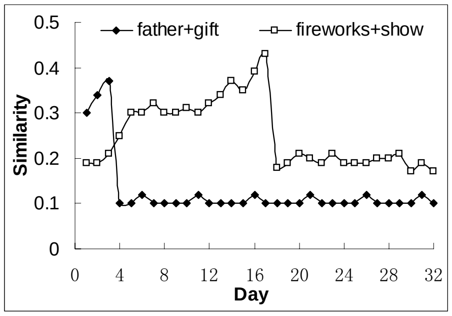

Figure 11: Daily-based query similarity evolution

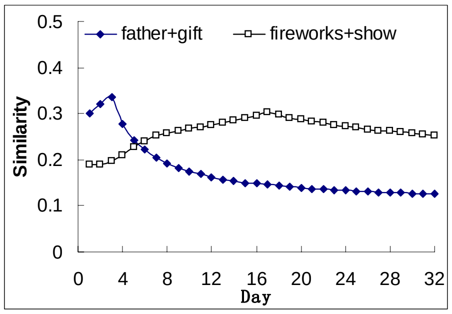

Figure 12: Query similarity with incremented approach

ond part. The reason is that we will use the first part as training data to construct the time-dependent query similarity model, while the second part is used to evaluate the model. Moreover, partitioning of the click-through dataset also depends on the user-defined *calendar schema* and *calendar patterns*. For example, to build a weekly based model, the training data should at least cover a time duration of one week; a yearly based time-dependent model cannot be constructed using click-through data of a few days.

Once the time-dependent query similarity model is constructed, given a query pair, the similarity can be obtained by matching the corresponding *calendar patterns* in the model. For example, with the weekly based query similarity model as shown in Figure 5, the query similarity between "kid" and "toy" can be derived based on the day of the week. We call the similarity derived from the model as *predicted similarity*.

Then, the predicted similarity value is compared with the exact similarity value calculated using the actual dataset. For example, with a weekly based similarity model constructed using the dataset in the first two weeks, the query similarity on the third Monday can be predicted, denoted as S'. Then, the exact similarity is calculated with the dataset in the third Monday. Given the predicted similarity value S' and the exact similarity value S, the *accuracy* of the model is defined as $\frac{|S-S'|}{S}$, where $|S-S'|$ is the absolute difference between the two values. Similarly, for the incremented approach, the same definition of accuracy is used so that we can compare the two approaches.

In the following experiments, a set of *1000* representative query pairs is selected from the query page pairs that have similarities larger than *0.3* in the entire click-through data. Some of them are the top queries in the week or month, some

$\|Trainingdata\|$	$\|Testingdata\|$	$Accuracy$
10	22	0.784
15	17	0.873
20	12	0.892
25	7	0.921
30	2	0.968

Table 2: Quality of the Time-dependent model (1)

are randomly selected, while others are selected manually based on the related real world events such as "father's day" and "hurricane". Note that the accuracy values shown below are the average accuracy values of all the testing query pairs.

4.4 Performance Evaluation

To evaluate the accuracy of the time-dependant query similarity model, three sets of experiments have been done. Firstly, the sizes of the data collections that are used to construct and test the time-dependant query term similarity model are varied. For example, we use the first twenty days as training data and use the eleven days left as testing data or we use the first thirty days as training data and use the last day left as testing data, etc. Note that as the size of the testing data increases, the distance between the training data and test data increases as well. Secondly, only the size of the data collection that is used to constructing the time-dependent model is varied. Whereas the testing data is always the other day followed the dataset being used. For example, we use the first twenty days as training data and use data in the 21^{st} day as testing data. Thirdly, the *distance* between the training data and testing data is varied while the sizes of the training data and testing data are fixed. Note that the *distance* between the two data collections is the distance between the latest query-page pairs in the two collections. For instance, we can use the first twenty days as training data and use data in the 21^{st} day as testing data for the case where *distance* is *1*. If the *distance* is *2*, then data in the 22^{nd} is used for testing. Note that all possible combinations of training and testing data that satisfy the distance constraint are used and the average accuracy values are presented. In the following experiments, if not specified, the calendar schema $< hour, day, month >$ is used with the calendar pattern $< 1, *, * >, < 1, *, * >, \cdots, < 24, *, * >$.

Table 2 shows the quality of the time-dependent query similarity model by varying the sizes of data that are used for constructing the model and testing the model. We can see that when the size of the training data increases and size of the testing data decreases, the accuracy of the time-dependent model increases as well. When the sizes of the training and testing data are similar, the accuracy can be as high as 87.3%. Note that here all the click-through data in the *32* days are used. We use the first part of the data as training data and the rest as testing data. The reason behind may be that when the training is not large enough to cover all the possible patterns, then the time-dependent model may not be able to produce accurate results.

Figure 13 shows the quality of the time-dependent query similarity model by varying the size of data that is used for construction the model and fixing the size of data that is used for testing to *1*. Three different calendar schemas and calendar patterns are used as well. We can see that when the size of the training data increases, the accuracy of the time-dependent model increases as well. This fact is just

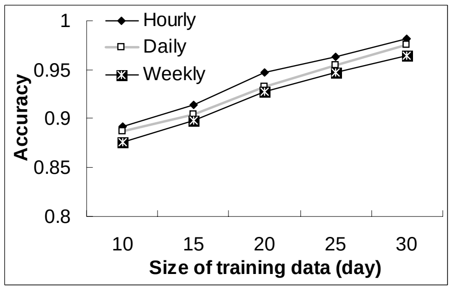

Figure 13: Quality of the Time-dependent model (2)

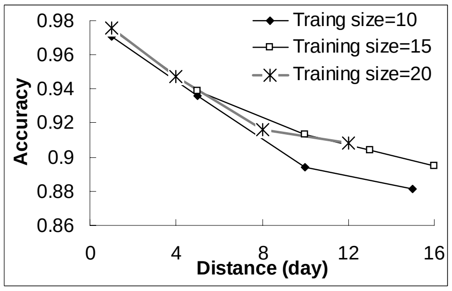

Figure 14: Quality of the Time-dependent model (3)

as we expected: as the size of the training data increases, performance of the model is expected to increase.

Figure 14 shows how the quality of the time-dependent query similarity model changes by varying the time distance between the data collection that is used for testing and the data collection used for constructing the model. For example, when the distance is *1* and the training data size is *10*, we summarize all the accuracy values that use the i to *10+i* days as training and use the *10+1+i* as testing. We can see that when the distance increases, the accuracy of the time-dependent model decreases. At the same time, when the size of the training data increases, with the same distance, the accuracy value may increase. The reason behind this set of data shows that the time-dependent model is more accurate if the most recent data is incorporated as the time-dependent model may be modified.

Moreover, we implemented an incremented query similarity model and compare the prediction accuracy with the time-dependant approach. Note that for the two approaches both the data that are used for building the model and the data that are used for testing are the same (The first part of the data is used for training and the rest is used for testing). In the following experiments, three calendar schema and calendar pattern pairs are used. The calendar schema and calendar patterns are $< hour, day, month >$ with $< 1, *, * >$, $< 2, *, * >$, $\cdots$, $< 24, *, * >$; $< hour, day, month >$ with $< *, 1, * >$, $< *, 2, * >$, $\cdots$, $< *, 31, * >$; and $< day, week >$ with $< 1, * >$, $< 2, * >$, $\cdots$, $< 7, * >$. We use the *1000* sampled query pairs for performance evaluation.

Figure 15 shows the comparison of quality about the similarity values obtained using the incremented approach and the time-dependent model. Note that the size of the training data is varied from 1/4 of the dataset to 7/8 of the dataset as well, while the rest is used for testing. We can see that when the intervals in the calendar schema become larger, the

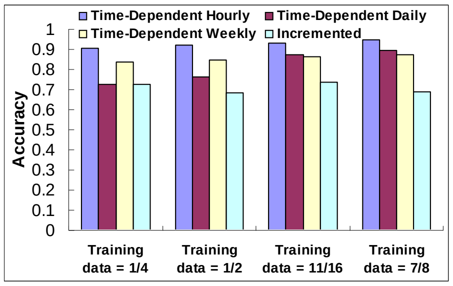

Figure 15: Quality of the Time-dependent model (4)

quality of the time-dependant model decreases. This is because we only use a click-through data of *32* days, which can produce satisfactory results with the hourly and daily based calendar patterns. Yet, the quality of the time-dependent model is generally better than the incremented approach. Moreover, we observe that for some calendar schema based time-dependent query similarity model, the accuracy of the model decreases dramatically when the size of the training data decreases, especially for the daily based calendar schema. The reason is that the size of our data collection is not large enough, thus when the size of the training data decreases it cannot cover every possible day in one month (requires at least 31 days of training data).

5. DISCUSSIONS AND FUTURE WORK

The experiments show that for most query pairs, the similarities are time-dependent and the time-dependent model can produce more accurate similarity values compared to the incremented approach. Besides the time dimension that affects the similarities between queries, there are other factors such as user groups, locations, and topic context, etc. In this paper, we have focused on incorporating the time dimension. In the future work, we will incorporate other factors mentioned above into the query similarity model. Two extended models are presented as follows.

Personalized time-dependent query similarity model: Beside the time dimension, user groups play an important role in determining the similarities between queries. This is based on the observation that different users have different search habits and have different query vocabularies [8]. For example, some users search about news and business information in the morning and entertainment information in the night, while others may have the reverse habits. Also, to describe the same object or event, people come from different background usually use different query terms. The personalized time-dependent query similarity model is to combine the user information together with the temporal information to build an accurate query similarity model that can be used for improving the personalized search experience.

Spatial-temporal-dependent query similarity model: Similar to the user groups, the spatial location [17, 18] of the queries and Web pages may affect the similarities between queries. For example, for the same object, users in the United States may have different information need compared to users in Asia. Also, contents of Web pages that are created by people in the United States may use different vocabularies compared to those created by people from Asia. By combining the spatial and temporal information, a spatial-temporal-dependent query similarity model can be constructed. As mentioned in [17], there are different types of locations such as *provider location*, *content location*, *serving location*, and *user location*. With such information, we believe, the spatial-temporal-dependent query similarity model can be used to improve the search experience.

6. CONCLUSIONS

With the availability of massive amount of click-through data in current commercial search engines, it becomes more and more important to exploit the click-through data for improving the performance of the search engines. This paper attempts to extract the semantic similarity information between queries by exploring the historical click-through data collected from the search engine. We realize that the correlation between query terms evolves from time to time in the click-through data, which is ignored in the existing approaches. Different from the previous work, we proposed a time-dependent semantic similarity model by studying the temporal information associated with the query terms in the click-through data. We formulated the time-dependent semantic similarity model into the format of kernel functions using the marginalized kernel technique, which can discover the explicit and implicit semantic similarities effectively. We conducted the experiments on the click-through data from a real-world commercial search engine in which promising results show that term similarity does evolve from time to time and our semantic similarity model is effective in modelling the similarity information between queries. Finally, we observed an interesting finding that the evolution of query similarity from time to time may reflect the evolution patterns and events happening in different time periods.

7. ACKNOWLEDGMENTS

The work described in this paper was partially supported by two grants, one from the Shun Hing Institute of Advanced Engineering, and the other from the Research Grants Council of Hong Kong S.A.R., China (Project No. CUHK4205/04E).

8. REFERENCES

[1] R. Attar and A. S. Fraenkel. Local feedback in full-text retrieval systems. *Journal of ACM*, 24(3):397–417, 1977.

[2] R. A. Baeza-Yates, R. Baeza-Yates, and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[3] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407–416, 2000.

[4] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized Web query log. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–328, 2004.

[5] S. Chien and N. Immorlica. Semantic similarity between search engine queries using temporal correlation. In *Proceedings of International World Wide Web Conference*, pages 2–11, 2005.

[6] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *Proceedings of International World Wide Web Conference*, pages 325–332, 2002.

[7] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[8] T. Heath, E. Motta, and M. Dzbor. Uses of contextual information to support online tasks. In *Proceedings of International World Wide Web Conference*, pages 1102–1103, 2005.

[9] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

[10] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 154–161, 2005.

[11] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of International Conference on Machine Learning*, pages 321–328, 2003.

[12] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 239–248, 2005.

[13] X. Shen, S. Dumais, and E. Horvitz. Analysis of topic dynamics in Web search. In *Proceedings of the International Conference on World Wide Web*, pages 1102–1103, 2005.

[14] J. Sun, H.-J. Zeng, H. Liu, Y.-C. Lu, and Z. Chen. Cubesvd: A novel approach to personalized Web search. In *Proceedings of the International World Wide Web Conference*, 2005.

[15] K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(1):268–275, 2002.

[16] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of ACM SIGMOD Conference*, pages 131–142, 2004.

[17] C. Wang, X. Xie, L. Wang, Y. Lu, and W.-Y. Ma. Web resource geographic location classification and detection. In *Proceedings of International World Wide Web Conference*, pages 1138–1139, 2005.

[18] L. Wang, C. Wang, X. Xie, J. Forman, Y. Lu, W.-Y. Ma, and Y. Li. Detecting dominant locations from search queries. In *International Conference on Machine Learning*, pages 321–328, 2003.

[19] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Clustering user queries of a search engine. In *Proceedings of the International World Wide Web Conference*, pages 162–168, 2001.

[20] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information System*, 18(1):79–112, 2000.

[21] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan. Optimizing Web search using Web click-through data. In *Proceedings of ACM CIKM Conference*, pages 118–126, 2004.