Understanding Temporal Intent of User Query Based on Time-Based Query Classification

Pengjie Ren, Zhumin Chen*, Xiaomeng Song, Bin Li, Haopeng Yang, and Jun Ma

School of Computer Science and Technology,
Shandong University, Jinan, 250101, China
chenzhumin@sdu.edu.cn

Abstract. Web queries are time sensitive which implies that user's intent for information changes over time. How to recognize temporal intents behind user queries is crucial towards improving the performance of search engines. However, to the best of our knowledge, this problem has not been studied in existing work. In this paper, we propose a time-based query classification approach to understand user's temporal intent automatically. We first analyzed the shared features of queries' temporal intent distributions. Then, we present a query taxonomy which group queries according to their temporal intents. Finally, for a new given query, we propose a machine learning method to decide its class in terms of its search frequency over time recorded in Web query logs. Experiments demonstrate that our approach can understand users' temporal intents effectively.

Keywords: Temporal Intent, Query Classification, Machine Learning.

1 Introduction

World Wide Web is a dynamic information space in which the number and content of pages continuously change over time. And, many queries could only be answered accurately under a specific temporal pattern. That is, queries are dynamic. When a user submits a query to a search engine, *Query's Temporal Intent* is the time of the target information which satisfies the user's needs. The temporal intent may include one/several time points or periods of time. And, it is dynamic and varies with time. A direct application of query's temporal intent is to provide search result pages for users more accurately by limiting these pages' publishing time belonging to the intent. In addition, search results can be grouped according to the multiple temporal intents. This can ensure the diversity of the search results. For example, a user specifying a query 'presidents cup' may need information related to one of many possible subtopics: the Presidents Cup in golf, chess, tennis, football etc., and they are belong to different temporal periods. Obviously, detecting all these subtopics by semantics is difficult. However, it

* Corresponding author.

G. Zhou et al. (Eds.): NLPCC 2013, CCIS 400, pp. 334–345, 2013.

is relatively easy to identify the query's temporal intents. Then, they can be utilized to improve or diversify the search results. Therefore, it is necessary to study a query temporal intent detection algorithm which can be used to discover a query's temporal intent automatically. However, due to queries submitted by users are usually short and ambiguous, as well as temporal intent is dynamics and its statistical properties of the target variable change over time in unforeseen ways, this problem is non-trivial.

In this paper, we propose a time-based query classification approach to try to detect user query's temporal intent automatically. We first analyze the shared features of queries' temporal intents distributions, such as full-time intent, most recent time intent, or burst time intent. And, these features can help to obtain some time-related latent semantics under queries. Based on this intuitional observation, we present a query taxonomy which group queries according to their temporal intents. Then, we observed that query's temporal intent can be detected from its search frequency distributions over time. Thus, for a new given query, we propose an algorithm to decide its class in terms of its search frequency curve recorded in Web query logs. The class of a query implicitly represents the user's temporal intent of her information need which can help to understand the query better. We have collected a large amount of queries from TREC (Text REtrievl Conference) and manually annotated their categories. Experimental results indicate that our time-based query classification algorithm can group queries effectively.

The rest of this paper is organized as follows. We introduce related work in Section 2. In Section 3, we present a query taxonomy which group queries according to their temporal intents. Section 4 gives our method of temporal intent based query classification. In section 5 we discuss the corresponding experiments. We make some conclusions and our future work in Section 6.

2 Related Work

There is a large amount of previous work on exploring temporal characteristics of Web queries. Zhou et al. [1] defined temporal intent variability as popularity changes between the subtopics of a single topic (query) over time. For a given query, they first calculated the probability of interest of each subtopic over its all subtopics. Then they used the mean of the standard deviation of each subtopic as the temporal intent variability of the query. Shokouhi [2] investigated seasonal query type which represent seasonal events repeat every year and initiate several temporal information needs. He focused on detecting seasonal queries using time-series analysis. He first decomposed a query's sequence into three components: level, trend and season. Then, if the decomposed season component and raw sequence have similar distributions, he classified the query as seasonal. [3] presented an approach for understanding the time-varying search query relationships which express commonality in user intent among multiple search queries at a given time. The time-varying query interactions reflect the changing user needs over some time period.

Zhang et al. took the temporal features of queries into consideration in query substitution for ad search [4]. They extracted temporal features from query frequency curves and proposed a novel temporal similarity measurement by integrating these new features with the query frequency distribution. Jones and Diaz in [5] pointed out that temporal properties of queries can be used to diagnose the quality of the retrieval. They presented three temporal classes of queries: atemporal query, temporally unambiguous query and temporally ambiguous query. Metzler et al. [6] investigated implicitly year qualified queries which is a query that does not actually contain a year, but yet the user may have implicitly formulated the query with a specific year in mind. Asuar et al. [7] studied temporal signatures of three different types of queries - Navigational, Adult and News queries, and proposed a method to classify a query into these three types by computing trends in query-clicks over time.

Chien and Immorlica utilized temporal correlation to identify sets of similar queries, suggesting that queries with similar frequency patterns are likely to be related [8]. They defined a formal metric for temporal similarity between queries and used it to mine sets of related queries from a search log. Nunes et al. [9] investigated the use of temporal expressions in Web queries. They found that temporal expressions are scarcely used in the queries. They also found that these expressions are more frequently used in certain topics such as Autos, Sports, News and Holidays. Dakka et al. in [10] proposed a framework for handling time-sensitive queries and automatically identify the important time intervals that are likely to be of interest for a query. Then, they built scoring techniques integrating the temporal aspect into the overall ranking mechanism.

Kira et al. [11] proposed a method to compute word relatedness using temporal semantics analysis. For a given word, they first represented it as a weighted vector of concepts extracted from concept repository such as Wikipedia or Flickr image tags and denoted by a time series. Then, they got two words semantic relatedness by computing the similarity of all possible concept pairs. Giuseppe et al. in [12] examined the correlation between relevance and time. Then, they proposed an approach exploiting the detection of publication time peaks for the query expansion in the Blog search domain. Kira et al. explored how to use time series technique to model and predict user behavior over time including trends, periodicities and surprises[13]. Jaewon and Jure explored temporal dynamics of online content[14]. They treated mentions or interactions with a particular piece of contents as a time series. Then, they proposed a k-means like algorithm which uses a special distance measure to cluster time series by their shape.

Although there is a growth in research investigating temporal characteristics of queries recently, to the best of our knowledge until now few work has been done to understand user query's temporal intent. The most similar work is that Anagha et al. analyzed the distribution of query popularity along four dimensions: the number of spikes, the shape of the spikes, the periodicity of the queries, and the overall trend in popularity [15]. However, most of them either focused on only one query type, or did not did not propose an approach to understand a query's temporal dynamics automatically.

3 Temporal Intent Based Query Taxonomy

Understanding queries temporal intent is fundamental to understanding the retrieval experience. In order to obtain some latent semantics from the distribution of queries' temporal intents, we first observe that queries' temporal intents include full-time intent, most recent time intent, burst time intent or periodic intent. Then, we discover that a query's temporal intent can be reflected by its search frequencies over time which can be seen as a time series. Finally, we group queries to the corresponding temporal intent classes according to their temporal characteristics reflected by their time series, as shown in Figure 1.

Bellow we present definitions of these query classes and corresponding search time series shapes in terms of their temporal intents.

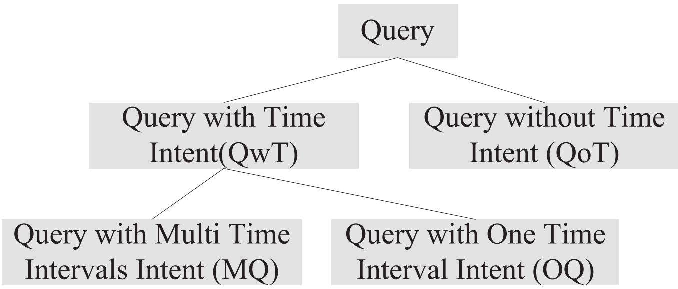


Fig. 1. Temporal Intent based Query Taxonomy

Query without Time Intent QoT denotes queries whose target information does not belong to any specific time. That is, there is no temporal constraint for their results. On the other hand, the temporal intent of QoT is full time. QoT denotes users' common, frequent and constant information needs. Consequently, their search frequency curves share a stable trend, for instance "Java JDK" as shown in Figure 2(a) derived from Google Trends [16].

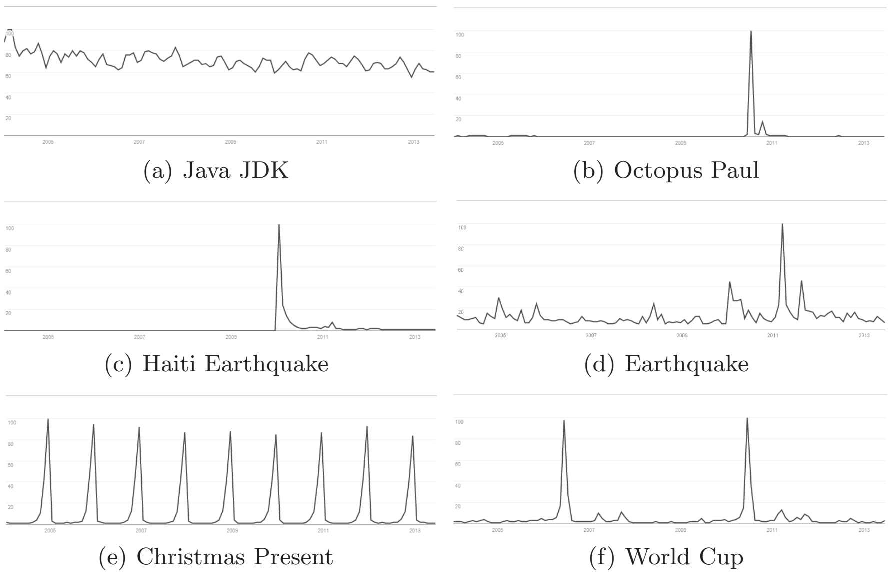
Query with Time Intent QwT denotes queries which contain implicit time intents.

Query with One Time Interval Intent OQ are these queries whose target information belongs to one specific time period. These queries are often triggered by one time unexpected event. As a result, their search curves all contain a single spike which occurs when there is a sudden increase followed by a corresponding decrease in query frequency. For example, Octopus Paul and Haiti Earthquake are OQ, as depicted in Figure 6(d) and 2(c) respectively.

Query with Multi Time Intervals Intent MQ are these queries whose target information belongs to multi time periods. These queries are often triggered by an event which repeated multi-times.

Query with Aperiodic Time Intervals Intent AMQ describes MQ whose multi time intervals are aperiodic. These queries are often triggered by an unexpected event or user requests, issued aperiodically. Search curves of AMQ share a common shape with multi aperiodic peaks. For example, "Earthquake" is a AMQ, as illustrated in Figure 2(d).

Query with Periodic Time Intervals Intent PMQ denotes MQ whose multi time intervals are periodic. These queries are often triggered by an expected event which follows identical or almost identical patterns during corresponding months of successive years. Search curves of PMQ share a common shape with multi periodic peaks. For example, "Christmas Present" gets hot in an annual cycle since it is time for people to select card for their friends in every Christmas, as shown in Figure 2(e). And, "World Cup" has the longer period of four years, as shown in Figure 2(f).



(a) Java JDK

(b) Octopus Paul

(c) Haiti Earthquake

(d) Earthquake

(e) Christmas Present

(f) World Cup

Fig. 2. Query Examples from Google Trends

4 Temporal Intent Based Query Classification

As mentioned above, we can see that the search frequency curves of these queries with the same temporal intent exhibit a common shape. And, queries with different temporal intents have different shapes. Therefore, we can understand queries' temporal intents by classifying queries into corresponding groups shown in Figure 1 according to their search frequency curves shapes.

Thus, the remaining problem is that for a new query, we propose a machine leaning algorithm to justify its category. To achieve this goal, the primary task is to compare the query data located at different positions of the time axis from Web query logs in order to detect pattern of the query. We use the conventional time series to represent temporal query data [17]. Let f_t denotes the frequency of query q issued by all users during the tth time interval, that 'month' is used here. $t = 1...N$ in which N is the number of time intervals. The frequency function F of a query q over N time intervals is a random f_t sequence, denoted as:

$$F = f_{t\ (t=1...N)} = \{f_1, f_2, ..., f_N\} \tag{1}$$

Preprocessing. F can be decomposed into three components [17], as shown in:

$$F = m_t + s_t + Y_t \tag{2}$$

Where m_t is a slowly changing function known as a trend component, s_t is a function with known period referred to as a seasonal component, and Y_t is a random, burst and irregular component. We first need to separate QoT and QwT, so we estimate and extract the s_t and Y_t. In other words, we remove m_t from F. Here we use Polynomial Fitting to get m_t [17],

$$m_t = \sum_{i=0}^{k} w_i x^i \tag{3}$$

In which k is set to 4 in this paper according the experiments. We choose the parameter w_i by minimizing the following target function:

$$L(W) = \sum_{t=1}^{N} (f_t - m_t)^2 + \frac{\lambda}{2}||W||^2 \tag{4}$$

Where $W = (w_0, w_1, w_2, w_3, w_4)$. After removing m_t, we get $F^q = s_t + Y_t$. An example is shown in Figure 3.

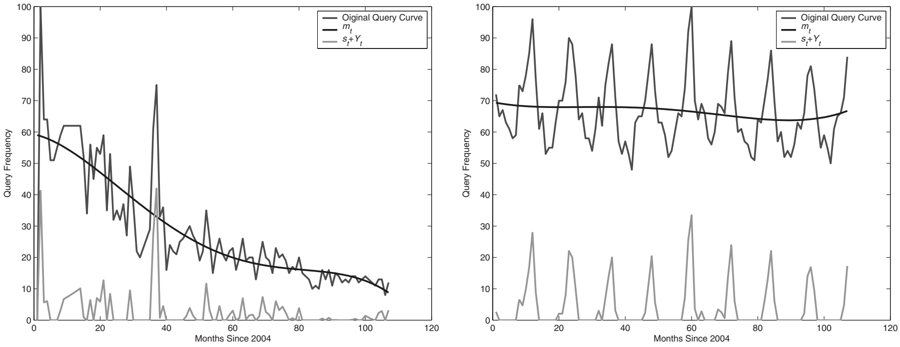
Features. We use 11 features for the machine learning model in this paper.

Feature1-2 QoT is stable while QwT is burst, so in order to distinguish QwT from QoT, Mean and Standard Deviation of F^q are two obvious features.

Feature3 However, it is difficult to separate OQ and MQ because all their Standard Deviations are larger. Hence we define a new feature as follows:

$$SR = \frac{f_M - \mathrm{Max}(\{f_1, f_2, ..., f_N\} - \{f_{M-m}, ..., f_{M-1}, f_M, f_{M+1}, ..., f_{M+m}\})}{\sum_{t=1}^{N} f_t} \tag{5}$$

In which, $f_M = \mathrm{Max}(\{f_1, f_2, ..., f_N\})$ is the frequency of the highest spike. $\{f_1, f_2, ..., f_N\}$ - $\{f_{M-m}, ..., f_{M-1}, f_M, f_{M+1}, ..., f_{M+m}\}$ denotes remove these points $\{f_{M-m}, ..., f_{M+m}\}$ from the set $\{f_1, f_2, ..., f_N\}$. m is a predefined parameter and $2m$ represents the duration of a spike. We determine m by analyzing these



(a) (b)

Fig. 3. Examples of Removing Trend Component

maximum spikes of all QwT in the query dataset [18]. First we define a threshold ratio r. If i and j satisfy all $\{f_{M-i}, ..., f_{M-1}, f_M, f_{M+1}, ..., f_{M+j}\} > r * f_M$, then $2m = i + j$. The analyzing result is shown in Figure 4. We can see that if $2m$=4, it can cover at least 74.6% queries regardless of the value of r. Thus, without loss of generality, in this paper we set $m = 2$.

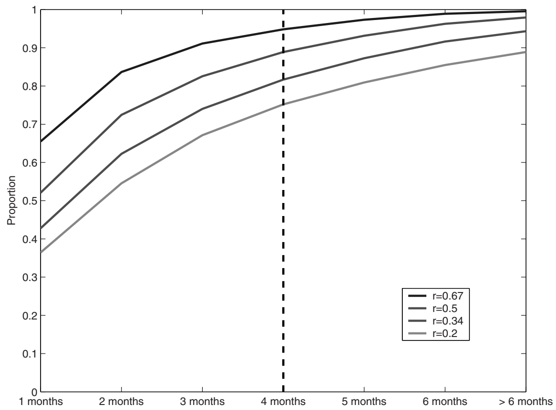


Fig. 4. Analyzing of Spike Duration

Feature4 Another feature is defined as:

$$MR = \frac{\text{Max}(\{f_1, f_2, ..., f_N\})}{\sum_{t=1}^{N} f_t} \tag{6}$$

Which represents the proportion of the maximum frequency and the sum of all frequencies.

Feature5-8 We adopt a distance measure that is invariant to scaling and translation of the time series [14]. Given two query curves $F1$ and $F2$, the distance $Distance(F_1, F_2)$ is defined as follows:

$$Distance(F1, F2) = \min_{\alpha,q} \frac{||F1 - \alpha F2_{(q)}||}{||x||} \tag{7}$$

where $F2_{(q)}$ is the result of shifting time series $F2$ by q time units, and $|| \cdot ||$ is the l_2 norm. This measure finds the optimal alignment (translation q) and the scaling coefficient α for matching the shapes of the two time series. With q fixed, $\frac{||x-\alpha y_{(q)}||}{||x||}$ is a convex function of α, and therefore we can find the optimal α by setting the gradient to zero: $\alpha = \frac{x^T y_{(q)}}{||y_{(q)}||^2}$. It is difficult to find the optimal q. In practice, we traverses all possible values of q to find out the minimum distance.

For a given query curve F, we compute its similarity to all cures of the other query categories in the training set. We use the mean similarity of the same query class as one feature. Then, we get four features, represented as D_{QoT}, D_{OQ}, D_{AMQ} and D_{PMQ}, corresponding to the query groups QoT, OQ, AMQ and PMQ respectively.

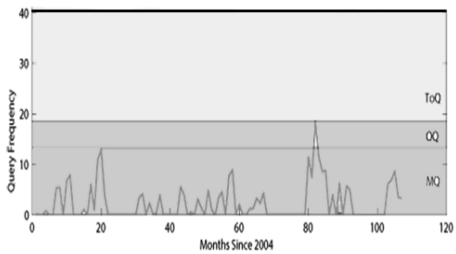


Fig. 5. Approximate *cutoff* of Training Data

Feature9-11 First we define the 9th feature *cutoff* as:

$$cutoff(X) : R^n - > R \tag{8}$$

Where R^n is the feature space. We need to learn *cutoff* from the training data. However, there are no annotated *cutoff* on ptraining data. So we have to get an approximate value of *cutoff* with Function 9 as shown in Figure 5.

$$cutoff = \begin{cases} \text{value of the median line of “yellow(first)” area} & if\ query = \text{QoT} \\ \text{value of the median line of “blue(second)” area} & if\ query = \text{OT} \\ \text{value of the median line of “pink(third)” area} & if\ query = \text{MT} \end{cases} \tag{9}$$

Then, we use the former 8 features *Feature 1-8* as the input of SVR (Support Vector Regression) [19] to get the *cutoff* of the testing data. In SVR, we use gaussian kernel function with model parameter $C = 22$.

The *cutoff* is used to detect spikes, and we define the number of these spikes as the 10th Feature. A spike is defined as some continuous points whose values are larger than *cutoff*.

In order to identify PMQ well, we define a new feature *PD*. According to *Feature10*, if there exist multi spikes, we use y_i to represent the time interval between two neighboring spikes as shown in Figure 3(b). We get a sequence $\{y_1, y_2, ..., y_w\}$. Then, *PD* is computed as the Standard Deviation of the sequence. Else if there is no or one spike, we set PD with extreme values.

5 Experiments

Corpora. For the lack of standard corpora for evaluating temporal intent based query classification algorithm, we have to construct data sets. We first extracted 5,000 queries from Web Track of TREC [18] and submit every query to Google Trends [16] and download its query frequency file. The numbers on the file reflect how many searches have been done for the particular query, relative to the total number of searches done on Google over time. We have to use the file as the corresponding query's frequency data to demonstrate our query classification algorithm because it is very difficult to get real and large-scale query logs from commercial search engines. Finally, we manually annotated categories of these queries in terms of their frequency curves and temporal intent based query taxonomy definitions described in Figure 1.

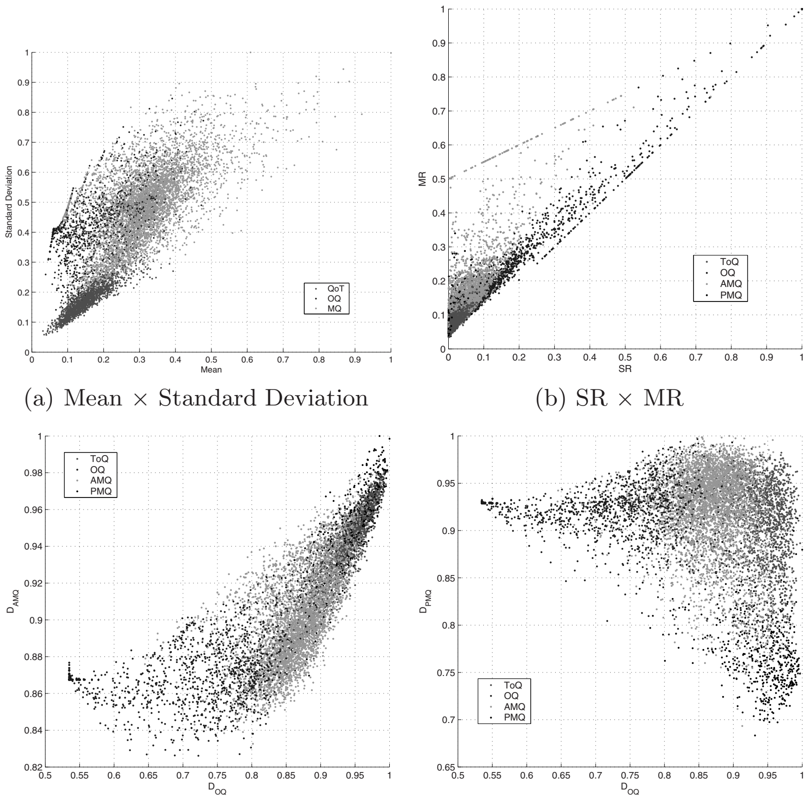
Evaluation Measures. We use *Precision* and *Recall* in evaluation of the temporal intent based query classification results. If the query category classified by the algorithm agrees with the manually annotated category, we view it as a correct classification. *Precision* is the fraction of classified query categories that are correct. *Recall* is the fraction of correct query categories that are classified. $F1$-*score* is calculated using following function: $F1 = 2 * (P * R)/(P + R)$.

Classifier and Parameter. With respect to the machine learning model, Support Vector Machine (SVM) [19] is used in this paper. The input are the eleven features and the output are the four categories. We used the C-Support Vector Classification in LIBSVM with the gaussian kernel function and set $C = 22$.

Table 1. Classification Performance Comparison for Different Query Categories

Query Class	QoT	OQ	AMQ	PMQ	average
P	0.952	0.928	0.846	0.914	0.910
R	0.973	0.915	0.831	0.924	0.911
$F1$	0.962	0.922	0.838	0.919	0.910

Results and Discussion. Table 1 shows the results. Because none of the previous approaches has provided an efficient method to group queries based on temporal intents, to the best of our knowledge, we have to only analyze our own approach. Obviously, our approach achieves high performance for all four query classes. We can see that the classification performance of AMQ is the worst. This is because queries tends to fluctuate caused by many factors and AMQ has more than one spike. If the fluctuation of the spike is not large enough, it is difficult to detect it. As a result, the query will be mistakenly classified as QoT. It is obvious the performance for QoT is the best among these four query classes for the reason that the Mean and Standard Deviations of all QoT's frequency curves are very low and our algorithm can identify it effectively. To our surprise, the performance of PMQ is also very high. This may because the feature SR and MR can distinguish it from the other query classes well.



(a) Mean × Standard Deviation

(b) SR × MR

(c) $D_{OQ} \times D_{AMQ}$

(d) $D_{OQ} \times D_{PMQ}$

Fig. 6. Feature Effect Analysis. Query Classes Distribution in Pair of Features Space

Feature Effect Analysis. We further analyze some typical features' effects by compute the query class distribution on feature space, as shown in Figure 6. It is obvious that the feature Mean and Stand-Deviation can distinguish QoT from QwT effectively as illustrated in Figure 6(a). The main reason for this is that

QoT's curves Means and especially Stand-Deviations are low. From the Figure 6(b), we can see that the feature combination of MR and SR can classify OQ and MQ well. MR is used to evaluate the proportion of the second spike frequency and the sum of all frequencies. As described in the figure, SRs of all MQ are high. As illustrated in Figure 6(c) and 6(d), the distance between queries of the same classes are small. This can be used to distinguish queries well.

6 Conclusion

In this paper, we study the problem of how to understand the implicit temporal intents of user queries. We propose a query classification method to solve this problem. We first analyze the temporal intents of Web queries. Then, we propose a query taxonomy based on queries' frequency over time. Finally, we introduce a machine learning method based on four features to classify queries into four categories. Experimental results demonstrate that our approach is effective.

In future work, we will explore more features for temporal intent based query classification. We also plan to explore the application of temporal intent. Especially, we will study how the temporal intent can be used to construct a page ranking model to improve information retrieval performance.

Acknowledgement. This work is supported by the Natural Science Foundation of China (61272240,61103151), the Doctoral Fund of Ministry of Education of China (20110131110028), the Natural Science foundation of Shandong province (ZR2012FM037) and the Excellent Middle-Aged and Youth Scientists of Shandong Province(BS2012DX017).

References

1. Zhou, K., Whiting, S., Jose, J.M., Lalmas, M.: The impact of temporal intent variability on diversity evaluation. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 820–823. Springer, Heidelberg (2013)
2. Shokouhi, M.: Detecting seasonal queries by time-series analysis. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, pp. 1171–1172. ACM. R, New York (2011)
3. Johansson, F., Färdig, T., Jethava, V., Marinov, S.: Intent-aware temporal query modeling for keyword suggestion. In: Proceedings of the 5th Ph.D. Workshop on Information and Knowledge, PIKM 2012, pp. 83–86. ACM, New York (2012)
4. Zhang, W., Yan, J., Yan, S., Liu, N., Chen, Z.: Temporal query substitution for ad search. In: SIGIR 2009, pp. 798–799 (2009)
5. Jones, R., Diaz, F.: Temporal profiles of queries. ACM Transactions on Information Systems 25(3), 14 (2007)
6. Metzler, D., Jones, R., Peng, F., Zhang, R.: Improving search relevance for implicitly temporal queries. In: SIGIR 2009, pp. 700–701 (2009)
7. Asur, S., Buehrer, G.: Temporal analysis of web search query-click data. In: SNA-KDD 2009, pp. 1–8. ACM (2009)

8. Chien, S., Immorlica, N.: Semantic similarity between search engine queries using temporal correlation. In: WWW 2005, pp. 2–11. ACM (2005)
9. Nunes, S., Ribeiro, C., David, G.: Use of Temporal Expressions in Web Search. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 580–584. Springer, Heidelberg (2008)
10. Dakka, W., Gravano, L., Ipeirotis, P.G.: Answering general time sensitive queries. In: CIKM 2008, pp. 1437–1438. ACM (2008)
11. Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S.: A word at a time: computing word relatedness using temporal semantic analysis. In: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, pp. 337–346. ACM, New York (2011)
12. Amodeo, G., Amati, G., Gambosi, G.: On relevance, time and query expansion. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 1973–1976. ACM, New York (2011)
13. Radinsky, K., Svore, K., Dumais, S., Teevan, J., Bocharov, A., Horvitz, E.: Modeling and predicting behavioral dynamics on the web. In: Proceedings of the 21st International Conference on World Wide Web, WWW 2012, pp. 599–608. ACM, New York (2012)
14. Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 177–186. ACM, New York (2011)
15. Kulkarni, A., Teevan, J., Svore, K.M., Dumais, S.T.: Understanding temporal query dynamics. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 167–176. ACM, New York (2011)
16. Google trend, `http://www.google.com/trends?hl=en`
17. Brockwell, P.J., Davis, R.A.: Introduction to time series and forecasting. Springer (2002)
18. TREC. Million query track, `http://trec.nist.gov/data/million.query.html`
19. Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)