Learning Temporal Ambiguity in Web Search Queries

Behrooz Mansouri
School of Electrical and Computer Engineering, University of Tehran
Iran
b.mansouri@ut.ac.ir

Mohammad Sadegh Zahedi
School of Electrical and Computer Engineering, University of Tehran
Iran
s.zahedi@ut.ac.ir

Maseud Rahgozar
Database Research Group, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran
Iran
rahgozar@ut.ac.ir

Farhad Oroumchian
Department of Computer Science and Engineering, University of Wollongong in Dubai, Dubai, UAE
farhadoroumchian@uowdubai.ac.ae

Ricardo Campos
Polytechnic Institute of Tomar
LIAAD - INESC TEC
Portugal
ricardo.campos@ipt.pt

ABSTRACT

Time has strong influence on web search. The temporal intent of the searcher adds an important dimension to the relevance judgments of web queries. However, lack of understanding their temporal requirements increases the ambiguity of the queries, turning retrieval effectiveness improvements into a complex task. In this paper, we propose an approach to classify web queries into four different categories considering their temporal ambiguity. For each query, we develop features from its search volumes and related queries using Google trends and its related top Wikipedia pages. Our experiment results show that these features can determine temporal ambiguity of a given query with high accuracy. We have demonstrated that a Multilayer Perceptron Networks can achieve better results in classifying temporal class of queries in comparison to other classifiers.

KEYWORDS

Temporal IR; Temporal Query Classification; Query Intent

1 INTRODUCTION

Many web search queries have implicit temporal intent associated with them [1]. Since web queries are short (only 2.5 terms on average), this creates ambiguity, with likely possible multiple interpretations. By way of example, consider the query “*FIFA World Cup*”, which may refer to different events that either took place (*2014, 2010, 2006)*, or are expected to occur in a near future (2018, 2022). The uncertainty to which one of the many time frames it refers to, makes this a temporal ambiguous query. Depending on the nature of the query, results of past, recency or future nature may be differently shown to the user. If the query is issued within the time-period of the World Cup event, then possibly the most recent pages concerning the event could be more relevant to the user’s needs. In contrast, in other time-periods, users might be eventually more interested in obtaining a snapshot of the results of the different world cup events. In this second scenario, one could expect an improved effectiveness if temporal diversity in retrieved web pages is applied.

To improve the search effectiveness of the results, detecting the temporal ambiguity of web search queries is of the utmost importance. One way to identify this temporal ambiguity is to use search engine query logs. Large-scale traffic logs offer a rich resource of temporal signals that may be useful to determine temporal intent behind a query [12]. Certain temporal patterns could be observed by examining search traffic variations over time. The temporal dynamics of web queries have been commonly studied by building time series for queries based on their past frequency at uniform intervals and extracting time series features [5, 9, 16]. An interesting tutorial on this topic has been given by Radinsky et al. [14]. Other than frequency volume, previous researches used click log, query reformulation and relevant documents to better understand user temporal intent [1, 3, 13, 18]. In particular, Jones and Diaz [9], introduce a model to measure the distribution of documents retrieved in response to a query over the time domain in order to create a temporal profile for a query. They introduced three temporal classes of queries: atemporal, temporally ambiguous and temporally unambiguous. Campos et al. [1] also propose to classify queries into one of these three categories using information extracted from web snippets. Metzler et al. [13] in turn, used query logs to investigate implicitly year qualified queries. To find these types of queries they investigate how strongly these queries are associated with several different years. Zhang et al. [18] focused in detecting recurrent queries that are about events which occur at predictable intervals. Shokouhi [16] used seasonality of query volume time series to detect seasonal queries. The work by Gupta and Berberich [5] describes a taxonomy of temporal classes at different granularities. Ghoreishi and Aixin [3] and Kanhabua et al. [10]. studied event-related queries within Temporralia task of NTCIR [8] which considers 4 classes: atemporal, past, present, and future. A fully detailed description on temporal information retrieval applications can be found in the survey of Campos et al. [2]. In this paper, we address the problem of learning temporal

CIKM'17 , November 6–10, 2017, Singapore, Singapore
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133129

ambiguity in web search queries. We introduce a new query classification scheme that groups queries into 4 different categories: temporal unambiguous queries (TU), unpredictable temporal ambiguous queries (UTA), predictable event-related temporal ambiguous queries (PERTA) and predictable commemorative temporal ambiguous queries (PCTA). Our work differs from previous approaches in that we focus on categorizing web queries by their temporal ambiguity. The relevant researches in this context are [5, 9]. In [9] time sensitive queries are divided into only two categories of temporal unambiguous and temporal ambiguous. [5]. proposed a taxonomy built on [9] where temporal ambiguous queries were divided into three subcategories by their granularity; ambiguous at day or month or year. Temporal ambiguous queries at year granularity were also categorized into periodic and aperiodic. However, in this research web queries are divided into four different categories by their temporal ambiguity and for each of these categories need an appropriate ranking approaches. In our work, in addition to time series features, we extract novel features from related queries and relevant Wikipedia pages. Table 1 shows each category with a few examples of its query instances. Ideally, search engines would have different retrieval strategies for any of the different categories, making use of this additional information to provide better responses for their users.

Table 1: Temporal Classes with Query Instances

Query Class	Query Instances
Temporal unambiguous (TU)	Computer Science, Secure passwords
Unpredictable temporal ambiguous (UTA)	Messi Injury, Tsunami, Tom Hanks movie, …
Predictable event-related temporal ambiguous (PERTA)	US presidential election, Golden globe awards, The US Open
Predictable commemorative temporal ambiguous (PCTA)	Santa Claus, Valentine's Day, September 11th, trick or treat, …

The remainder of this paper is organized as follows. Section 2 introduces our temporal classification taxonomy. Section 3 describes the features used for classification. Section 4 outlines our experiment results. Finally, Section 5 provides some conclusions.

2 TEMPORAL CLASSIFICATION TAXONOMY

In this section, we introduce our new classification scheme. We classified the queries into four categories. (1) Temporal unambiguous queries (TU) have no or only a single time interval. Their query volume shows either no specific pattern or a pattern with only one spike. For example, the time series associated with the search volume of the query "*Computer Science*" looks like Figure 1. The content relevance is the most appropriate ranking strategy for this type of queries. (2) The unpredictable temporal ambiguous queries (UTA) have multiple time of interest (spikes) but these time intervals show no specific pattern. Figure 1 shows search frequency of query "*Tsunami*". For these types of queries, search engine can implement two different strategies. If the query is now trending, then more weight should be given to recent documents. For non-trending (normal) cases, temporal diversity in the results can improve ranking. Often, users prefer pages like Wikipedia to first understand what the phenomena is and then pages of different times to understand what happened during each time interval. (3) The predictable event-related temporal ambiguous queries (PERTA) are associated with events that happen within regular time intervals but each episode is independent of the others. For instance, "*Cannes film festival*" is an annual film festival but each edition is different from the others. Figure 1 demonstrates the periodic pattern of searches for this event. During the peak times, users prefer to know about the current edition while on other occasions, they prefer a temporal diversity in the result set. The last category is (4) predictable commemorative temporal ambiguous (PCTA) which contains queries related to commemorative days and seasonal activities. The pattern of the volume search time series for this type is similar to the previous group. Figure 1 shows search volume of query "*Trick or treat*". The difference between this type and the previous one is in that the periodic events of this type of queries are not completely different from each other. For this type of queries, users will likely prefer to start seeing pages like Wikipedia that gives them the basic knowledge. However, when the query is issued during the peak times, some users may likely prefer recent pages. For example, for the query "*Halloween costume*", people may want information about where they can buy costumes and recent documents are more appropriate in this case.

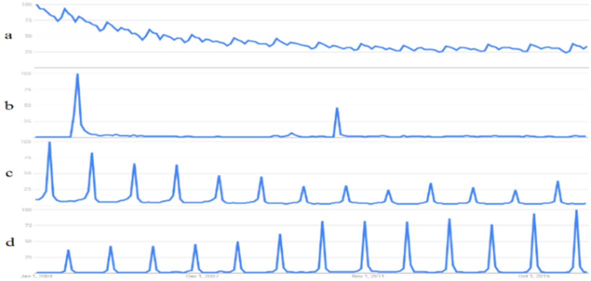

Figure 1: Query temporal pattern examples from Google Trends from January 2004 to January 2017 (a: "*computer science*" – TU query, b: "*Tsunami*" – UTA query, c: "*Cannes film festival*" – PERTA query, d: "*Trick or treat*" – PCTA query). The horizontal axes represent the time while the vertical axes show query search volume.

3 OUR APPROACH

To detect temporal ambiguous queries, we expand the queries with three types of features: (i) time-series features (ii) related-queries features (iii) Wikipedia features.

3.1 Time Series Features

A time series is a sequence of values of a particular measure taken at regularly spaced intervals over time. For each query in the context of web search, a time series can be generated using its relevant documents published time or its past frequency at uniform intervals. In this work, we chose the latter. Here we introduce our 7-time series features:

- **Autocorrelation** indicates how well a time series is similar to a time-shifted copy of itself. We used lag-1 autocorrelation of a time series which is the correlation of each value with the immediately preceding observation. Time series of queries with strong inter-day dependency have higher autocorrelation value [10] Autocorrelation of time series T with lag=1 can be calculated as follows ($\bar{t}$ is the mean value of time series):

$$Autocorrelation(T) = \frac{\sum_{i=1}^{N-1}(t_i - \bar{t})(t_{i+1} - \bar{t})}{\sum_{i=1}^{N}(t_i - \bar{t})} \qquad (1)$$

- **Seasonality** represents the cosine similarity between time series and its seasonal component. This feature was introduced by [16] to detect seasonal queries. In this work, we use Holt-Winters decomposition technique [4]. After decomposing the time series, we remove its trend component from the time series as it just shows the overall trend of a query and then calculate the cosine similarity between the seasonal component and the remaining components. Considering S as seasonal component of time series and $\hat{T}$ as time series with removed trend component seasonality of time series T is:

$$Seasonality(T) = \frac{S\,.\hat{T}}{\|S\|\,.\,\|\hat{T}\|} \tag{2}$$

- **Kurtosis** calculates how much of the probability distribution is contained in the peaks and how much in the low-probability regions [9]. and is calculated as the ratio of the fourth moment and variance squared.
- **Randomness Test** is used to analyze the distribution of a set of data to see if it is random. We calculate p-value of Mann-Kendall rank test [11]. and use it as a feature of randomness.
- **SSE** (Sum of Squared Errors) of a prediction model can show how the time series is unplanned at a given point. We estimate predicted values using Holt-Winters [4] approach.
- **Modality** in time series show number of detected modes. Temporal ambiguous queries should have multi-modal time series. In our work, we used Dip test [6] to calculate number of modes.
- **Mean** value of time series.

3.2 Related Queries Features

We used query log and related queries similar to [13]. In this research, the authors considered association of a query with a given year and calculated the number of times that the base query was used within that year. If a query had association with more than one unique year it was implicitly year qualified. In contrast, we consider in our work, related queries, which refer to the most frequent searches within the same user's search session. Considering related queries, we extract two features: the ratio of related queries containing a year and the number of total related queries. The second feature is the number of unique years mentioned in them. For instance, for the query "*Olympics*" and the related queries "*Summer Olympics*", "*2016 Olympics*" and "*2012 London Olympics*" the value for the first and second features would be 0.66 and 2 respectively. In order to detect year expressions, we consider any 4-digit numbers between 1800 and 2100 a year expression. Given the simplicity of the task we opt not to use a temporal tagger.

3.3 Wikipedia Features

Wikipedia is a free encyclopedia, written collaboratively by the people who use it. Several researches used Wikipedia data [7, 17, 19]. In this research, each query was issued to Wikipedia search and the number of year expressions in the name of top related pages was considered as a feature. This feature differs from the number of year expressions in related queries. In Wikipedia, the year expression indicates a real event while a year expression in query logs does not necessary mean the same. For example, in query logs we can find a query "*Halloween 2016*" which may be issued by users to see for example where they can buy a costume. But in Wikipedia no page exists for this query. On the other hand, for a query "*2016 Summer Olympics*" a page exists on Wikipedia which indicates it was a real event that happened in *2016*.

4 EXPERIMENTS

4.1 Dataset and Experimental Setting

Providing relevant query dataset for a temporal query classification task is challenging. Thus, by considering suitable queries from [5, 9] and manually compiling some web queries, our experiments were conducted on 500 queries manually labeled by 3 professional editors. An inter-rater reliability analysis using the Fleiss Kappa statistics was performed to determine consistency among the editors. Overall, the annotators obtained about 0.74 of agreement level, which represents a high agreement between editors. Each query was issued to Google trends and their search frequency volume and related queries were downloaded. The related queries for a query, are terms that are most frequently searched with its terms in the same search sessions. We set the time range between January 2004 (which marks the start of Google Trends) and January 2017. Table 2 summarizes the queries dataset. We also issued queries to Wikipedia to get the name of top-20 related pages. Our dataset is publicly available[1]. We used 10-fold stratified cross validation, and averaged the results over 10 runs. We used multilayer perceptron neural network (MLP) for the classification. MLP utilizes a supervised learning technique called backpropagation for training the network [15]. We designed a multilayer perceptron network having a single layer of 10 hidden units using a learning rate of 0.3 and a momentum term 0.2. We compared this classifier with LibSVM, Random Forest, AdaBoost, and Naïve Bayes. The experiments have been carried out using weka 3.8.0.

Table 2: Queries collections summary

Query Class	#Queries	Query Class	#Queries
TU	185	PERTA	100
UTA	105	PCTA	110

4.2 Feature Evaluation

In order to study the importance of our features we used information gain ratio (IGR) on training data to study the relevance of features to the classification. Table 3 list the features ranked by their information gain ratio. As it can be seen the most informative feature is extracted from related queries and the least important ones are extracted from time series.

Table 3: Features ranked by the information gain ratio

Rank	Feature	IGR
1	The ratio of related queries with year expression and total related queries	0.176
2	Mean of query search volume	0.142
3	Autocorrelation	0.140
4	Year expressions in title of Wikipedia pages	0.137
5	Unique year expressions in related queries	0.125
6	Sum of squared errors	0.116
7	Modality of time series	0.108
8	Kurtosis	0.092
9	Seasonality	0.046
10	Randomness of time series	0.005

[1] http://dbrg.ut.ac.ir/TemporalAmbiguousQueryDataset/TemporalAmbiguousQueryDataset.rar

4.3 Experimental Results

For the task of evaluating our proposal, we compared MLP classifier with four baselines: LibSVM, Random Forest, AdaBoost and Naïve Bayes. The results obtained are shown in Table 4. All three measures were calculated as weighted average over all classes. As shown in this table, MLP classifier outperforms the other classifiers while the worst performance is for Naïve Bayes classifier. All the results are statistically significant when comparing Random Forest classifier with the other classifiers with p-value < 0.05 using the matched paired one-sided t-test.

Table 4: Performance of different classifiers

Model	Precision	Recall	F-measure
MLP	0.868	0.868	0.867
LibSVM	0.773	0.766	0.758
Random Forest	0.815	0.820	0.817
AdaBoost	0.793	0.836	0.814
Naïve Bayes	0.788	0.790	0.788

4.4 Failure Analysis

In our approach, the highest F-measure belongs to temporal unambiguous (TU) category at 0.915. On the other hand, the worst precision belongs to predictable commemorative queries (PCTA) at 0.816 while the worst recall is for predictable event-related queries (PERTA) with 0.75. To better analyze the reason for some failures of our proposed approach, we provide the confusion matrix for the MLP classifier in Table 5.

Table 5: Confusion matrix for the MLP classifier

Classified Real	TU	UTA	PERTA	PCTA
TU	174	7	2	2
UTA	13	87	3	2
PERTA	4	3	75	18
PCTA	4	2	6	98

As this table shows, some instances of predictable event-related query (PERTA) category were wrongly labeled as predictable commemorative queries (PCTA). As mentioned in section 2, query frequency volume for both of these categories have a seasonal pattern. Based on these results we can conclude that time series features cannot differentiate these two types of categories. The main reason for this misclassification is the lack of Wikipedia pages with year expressions for these types of queries. For example, for query “*Summer Camp*” (a supervised program for children during the summer months) no Wikipedia page exists. A further possible explanation to the misclassification of these type of queries, may be related to the simplicity of our assumption in just considering only year expressions in the title of related Wikipedia pages. For instance, a query like “*Super Bowl*” has different Wikipedia pages for each year of the event but they are mentioned with Latin numerals for example “*Super Bowl XXXVIII*”. Furthermore, as it can be seen from Table 5, some instances of unpredictable temporal ambiguous queries (UTA) were wrongly classified as temporal unambiguous (TU). This was mostly due to random shape of the time series. For example, for the query “*Bank robbery*” (a temporal unambiguous query), search frequency volume is as in Figure 2, which seems to be stable without high peaks while time series for this type of queries is expected to have non-periodic high peaks.

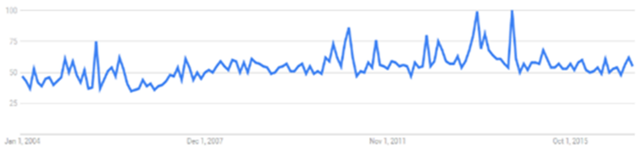

Figure 2: Search frequency volume for query “Bank robbery”.

5 CONCLUSIONS

In this paper, we proposed an approach for identifying different types of temporal ambiguous queries. We extracted features from search frequency volume and related queries using Google trends data and expanded our queries with these features. As a further additional knowledge, we also used top related Wikipedia pages title in order to extract year expressions for each query. A Multilayer perceptron neural network was used for temporal classification of queries. We have demonstrated that a reasonably good accuracy could be achieved for most of the categories. In future work, we plan to improve our categorization techniques by employing more distinctive features within a web retrieval search engine and by using standard test collections for experimental procedures.

ACKNOWLEDGMENTS

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project «POCI-01-0145-FEDER-006961», and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

REFERENCES

[1] Campos, R., Dias, G., and Jorge, A. (2011). What is the Temporal Value of Web Snippets. In WWW-TWAW’11, pp. 9-16.

[2] Campos, R., Dias, G., Jorge, A., and Jatowt, A. (2014). Survey of Temporal Information Retrieval and Related Applications. In CSUR, 47(2). Article N 15.

[3] Ghoreishi, S., and Aixin, S. (2013). Predicting Event-Relatedness of Popular Queries. In CIKM’13, pp. 1193-1196.

[4] Goodwin, P. (2010). The Holt-Winters Approach to Exponential Smoothing: 50 Years Old and Going Strong. In Foresight: The International Journal of Applied Forecasting, 19, pp. 30-33.

[5] Gupta D. and Berberich, K (2015). Temporal Query Classification at Different Granularities. In SPIRE’15, pp. 156-164.

[6] Hartigan, J. A., and Hartigan, P.M. (1985). The Dip Test of Unimodality. In The Annals of Statistics, 13(1), pp 70-84.

[7] Hu, J., Wang, G., Lochovsky, F., Sun, J-T., and Chen,m Z. (2009). Understanding User's Query Intent with Wikipedia. In WWW’09, pp. 471-480.

[8] Joho, H., Jatowt, A., Blanco, R., Naka, H., and Yamamoto, S. (2011). In NTCIR’11.

[9] Jones R. and Diaz, F (2007). Temporal Profiles of Queries. In TOIS, 25(3).

[10] Kanhabua, N., Nguyen, T., and Wolfgang, N. (2015). Learning to Detect Event-Related Queries for Web Search. In WWW’15, pp. 1139-1344.

[11] Kendall, M. G. (1948). Rank Correlation Methods.

[12] Kulkarni, A., Teevan, J., Svore, K. M., and Dumais, S.T. (2011). Understanding Temporal Query Dynamics. In WSDM’11, pp 167-176.

[13] Metzler, D., Jones, R., Peng, F., and Zhang, R. (2009). Improving Search Relevance for Implicitly Temporal Queries. In SIGIR’09, pp. 700-701.

[14] Radinsky, K., Diaz, F., Dumais, S., Shokouhi, M., Dong, A., and Chang, Y. (2013). Temporal Web Dynamics and its Application to Information Retrieval”. In WSDM’13, pp. 781-782.

[15] Rumelhart, D. E., Geoffrey E. H., and Ronald J. W. (1985). Learning Internal Representations by Error Propagation. In Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1, pp. 318-362.

[16] Shokouhi M (2011). Detecting Seasonal Queries by Time-Series Analysis. In SIGIR’11, pp. 1171-1172.

[17] Sil, A., and Silviu, C. (2014). Towards Temporal Scoping of Relational Facts based on Wikipedia Data. In CoNLL’14. pp. 109-118.

[18] Zhang, R., Konda, Y.; Dong, A.; Kolari, P.; Chang, Y., and Zheng, Z. (2010). Learning Recurrent Event Queries for Web Search. In EMNLP’10, 1129-1139.

[19] Zhao, Y., and Claudia H. (2016). Temporal Query Intent Disambiguation using Time-Series Data. In SIGIR’16, pp 1017-1020.