Query Ambiguity Identification Based on User Behavior Information

Cheng Luo^{*}, Yiqun Liu, Min Zhang, and Shaoping Ma

State Key Laboratory of Intelligent Technology and Systems Tsinghua National Laboratory for Information Science and Technology Department of Computer Science and Technology, Tsinghua University Beijing 100084, China c-luo12@mails.tsinghua.edu.cn,{yiqunliu,z-m,msp}@tsinghua.edu.cn

http://www.thuir.cn

Abstract. Query ambiguity identification is of vital importance for Web search related studies such as personalized search or diversified ranking. Different from existing solutions which usually require a supervised topic classification process, we propose a query ambiguity identification framework which takes user behavior features collected from click-through logs into consideration. Especially, besides the features collected from query level, we focus on how to tell the differences between clear and ambiguous queries via features extracted from multi-query sessions. Inspired by recent progresses in word representation researches, we propose a query representation approach named "query2vec" which constructs representations from the distributions of queries in query log context. Experiment results based on large scale commercial search engine logs show effectiveness of the proposed framework as well as the corresponding representation approach.

1 Introduction

Traditional search engines usually focus on satisfying elaborative information needs. However, research [1] shows that queries often come out ambiguous or underspecified, especially with the increase of mobile device usage in recent years.

When a search engine receives an ambiguous/broad query from a user and it knows little about this user, the best it can do is to present a diversified result list that covers several interpretations/subtopics of the query (see Figure 1 for an example). On the contrary, if the search engine knows much about the user, it can generate a personalized search result list according to the user's profile [2]. Compared with traditional search tasks, diversified search and personalized search tasks require more computational resources. Meanwhile, diversification may lack effectiveness, and may even hurt the performance of clear queries. Therefore, query ambiguity identification is essential for assessing whether search

^{*} This work was supported by Natural Science Foundation (60903107, 61073071) and National High Technology Research and Development (863) Program (2011AA01A205) of China.

2 Query Ambiguity Identification Based on User Behavior Information

engine should introduce a diversified ranking or personalized search process. However, identifying a query's ambiguity manually is laborious and complicated. When judging a query, a human annotator usually makes a decision based on his/her personal experience. In such case, the annotator might not be able to make the correct judgment until he/she invests enough time in researching on a specific query. It is thus of vital importance for Web search engines to identify query ambiguity automatically.



Fig. 1. Search Result Page for an Ambiguous Query "cherry"

To identify ambiguous queries automatically, there have been a number of works such as [3,4]. They classified queries into 3 categories: Ambiguous, Broad and Clear. Based on this taxonomy, they proposed a supervised learning model to classify queries to Type A (Ambiguous) and Type \overline{A} (Broad and Clear). The top retrieved documents are crawled as a description of the query. Each document is classified into predefined categories and modeled as a possibility distribution on different topics. Further, they estimate the distances between documents and train a binary classifier with Support Vector Machine.

We can see that, to a great extent, most existing query ambiguity identification frameworks depend on a process of supervised document topic classification. It is an ambitious move to construct a high-quality training corpus for document classification. More importantly, the training corpus should be updated with the rapid changes of Web content, which is both time/labor consuming and almost impossible for practical Web search applications.

In this work, we try to address the problem of query ambiguity identification without topic-based classification. We propose a query ambiguity identification framework which adopts a supervised learning approach to automatically identify query ambiguity based on the features extracted from a number of evidences, i.e., query logs, SERPs and query sessions. Inspired by recent word representation approach "word2vec" [5], we adopt the algorithm on large scale of search sessions to present queries as vectors and then extract features for ambiguity identification.

Our main contribution includes: (1) we proposed a query ambiguity identification framework without a content classification; (2) we proposed a novel query ambiguity description method based on query representation, to the best of our knowledge, this is the first approach that adopts "word2vec" methods in the context of Web search user behavior; (3) we construct a dataset for query ambiguity identification with information collected from a practical search engines.

2 Related Work

2.1 Query Ambiguity Taxonomy

Most of the previous query intent identification approaches were based on a famous taxonomy of 3 categories: informational, navigational and transactional; this approach was proposed by Border et al. and Rose et al. [7,8]. To judge if search results benefits from diversification, query ambiguity categories is more instructive for search engines.

Song et al. reviewed previous works[9, 10] and constructed the taxonomy of queries from being ambiguous to specific in three types: **Ambiguous**: a query that has more than one meaning; **Broad**: a query that covers a variety of subtopics, and a user might look for one of the subtopics by issuing another query; **Clear**: a query that has a specific meaning and covers a narrow topic [3]. A user study on 60 queries which five human subjects involved in indicates that human annotators' judgment reach an agreement as high as 90% when classifying queries into Type A and Type \overline{A} (Type B and Type C), while it is more difficult for the annotators to discriminate whether a query falls under Type B or Type C.

Another taxonomy about query ambiguity differentiate queries by how "narrow" or "general" the intent of the query reflects [11]. Narrow queries usually means users submit queries for specific answers, such as "how to configure java", while general queries means that users attempt to visit some relevant web resources like the query "New York".

2.2 Query Classification

Web queries can be classified according to different taxonomies, such as search task type (Navigational, Informational and Transactional), query topics (KDD-Cup 2005) [12] and query ambiguity (Ambiguity vs Broad vs Clear, or Narrow vs General).

The search result lists returned by search engines are often regarded as a pseudo-relevance feedback [13]. The KDD-CUP participants started the trend of using the search engine directory services and the categories of the open directory project (ODP) to classify queries to predefined categories [14]. Document content, anchor links, hyperlinks and URLs have been considerably used in classifications. The importance of inlinks, authorities, hubs and anchors in web query classification were compared by Westerveld et.al [15].

Several previous works included user behavior as a functional feature in query classification [16, 17]. Liu et al. proposed a method to differentiate navigational queries from informational/transactional queries with features extracted from

user click histories [18]. Wang et al. [19] proposed an ambiguous query identification method with user entropy.

Song et al. have proposed a SVM classifier to solve the query ambiguity classification problem [3]. In their approach, each query is represented by possibility distributions on content categories of its top retrieved documents. Features are extracted from the distances between the possibility vectors. In a following work, they included user clicks which may affect the weights of document [4], and improves the classification performance.

However, as stated earlier, document classification is a great challenge on web environment because the construction and update of training corpus is complex and time-consuming. It must be pointed out that there is a significant difference between Song et al.'s approach and ours in the way of query session information utilization. They scrawled the top 20 snippets of follow up queries for content classification while we attempt to analyze the distributions of vectors whose queries are in session context.

Hafernik et al. proposed a less expensive method to classify queries into narrow or general using only part of speech (POS) information and query length as features [11]. It could be argued they did not utilize any user behavior information for classification.

2.3 Distributional Representation

Distributional representation has been successfully used for a wide variety of tasks in natural language processing and information retrieval, such as synonym detection, concept categorization and sentiment analysis. Mikolov et al. proposed a skip-gram based method significantly improved the training speed and vector quality by subsampling of frequent words [5, 6]. The linearity of skip-gram model make it very suitable on process large amount of documents.

3 Methods

4

To assess diversification or personalization opportunities, we are focusing on differentiating queries that reflect clear user intents between the ones that needs diversification. We adopt the taxonomy and ambiguity definitions proposed by Song et al., however, we attempt to classify query into Type C (Clear) and Type \overline{C} (Ambiguous and Broad). This is because ambiguous queries and broad queries need diversified search or personalized search while clear queries need not, which means we have to take broad queries into account while deciding whether diversification or personalization is needed. Thus, from this perspective, classification between Type C and Type \overline{C} is more instructive for search engines.

3.1 Problem Formulation and Framework Overview

The problem of query ambiguity identification can be formulated as a classification process:

$$f: Q \vdash \{C, C\} \tag{1}$$

f is the identification function which indicates whether the query is clear or not and $Q = \{q_1, q_2, q_3, \ldots\}$, which denotes the query space,

The framework is shown in Figure 2. It is constructed on three levels: data, feature and application.



Fig. 2. Query ambiguity identification framework

On the data level, given a query q_i , we first enrich the information on q_i with various evidence resources. By scrawling search result pages of the query, we can get the most relevant documents about the query. From user click-through data, we can learn the interaction pattern of specific users. By collecting large scale of user sessions, we can adopt query representation algorithm on the dataset to present every query as a vector for further analysis.

On the feature level, we extract features from both the query q_i itself (for example, the query length, if the query contains stopwords etc.) and the information enriched before. In this work, we investigate the effectiveness of **Basic Features (BF)**, User Click Behaviors (UCB), Search Result Hints (SRH) and Session Features (SF).

On the application level, we construct a training set by sampling some queries from search logs and labeling them by ambiguity. Each query and its label can be presented as (q_i, l_i) where q_i is a query and the label $l_i \in \{-1, +1\}$. We apply Support Vector Machine (SVM) to learning a classifier that minimizes the generalization error, or at least an upper bound on it. We select the nonlinear SVM with RBF (Radial Basis Function) kernel as our setting because it performs well on our dataset.

3.2 Query Enrichment and Feature Extraction

In this section, we describe how to use user clicks, sessions and SERPs to enrich a query and extract features that are designed to distinguish query ambiguity.

Basic Features There are three basic features that may be useful in identifying query ambiguity: (1) **QueryLength**: the length of the query; (2) **IfContainStopword**: if the query contains a stopword; (3) **IfContainQuestion**: if the query contains a question or interrogative. All of the three features can be 6 Query Ambiguity Identification Based on User Behavior Information

extracted without any other web resource. It is notable that *QueryLength* is assumed to be very helpful because usually the longer the query is, the more intent information it may carry. If a query contains a stopword or appears to be a question, it usually describes the user's intent specifically. Thus, *IfContainStopword* and *IfContainQuestion* are supposed to be functional.

User Click Features Previous works shows that user click features are very helpful in query intent classification and they might be functional to our query ambiguity identification task, because query intent in some degree suggests the ambiguity of query. For example, a navigational query usually reflects clear intent to visit the homepage of a website.

Liu et al. proposed two assumptions in [18]: Less Effort Assumption (user tends to click a small number of URLs when performing a navigational type search) and Cover Page Assumption (user tends to click only the first few URLs when performing a navigational type search).

Based on these two assumptions, they proposed two features nClickSatis-fied(nCS) and nResultSatisfied(nRS) for query intent classification as follows :

$$nCS = \frac{\# (Sessions \ of \ q \ that \ involves \ less \ than \ n \ clicks)}{\# (Sessions \ of \ q)}$$
(2)

$$nRS = \frac{\#(Sessions \ of \ q \ where \ only \ top \ n \ results \ are \ clicked)}{\#(Sessions \ of \ q)}$$
(3)

Click distribution proposed by Lee [20] is believed to be functional. We calculate the percentage of clicks on the top-n clicked results (nTCP).

$$nTCP = \frac{\sum_{i=1}^{N} \# \left(Clicks \text{ on } i^{th} \text{ most } freq \text{ clicked } result \right)}{\# \left(Clicks \text{ in session } of \text{ } q \right)}$$
(4)

Besides these features, we are introducing two features which indicates the types of user clicked result, RecommClickedPercentage(RCP) and CommonRe-sultClickedPercentage(CRCP) defined as follows:

$$RCP/CRCP = \frac{\#(Clicks \ on \ recommendation/common \ results \ on \ SERP \ of \ q)}{\#(Clicks \ in \ session \ of \ q)}$$
(5)

The type distribution of user-clicked results might suggest some characteristics of the query specificity. For example, if many users submit a query and then click its query recommendation on the SERP, this process of actions may suggests that it is very possible that the query is ambiguous and user needs to narrow down to a subtopic with the help of recommendations.

Search Result Hints Modern search engines usually incorporate search results from multiple vertical search engines into SERP (See an example in the Figure 3). These vertical search engines usually focus on some categories of web resource

such as videos, images, maps and dramas etc. When the search engine receives a query, it might activate some vertical search engines, and vertical search results might be incorporated into the SERP according the result of ranking algorithm. This way, we can try to figure out what the queries are talking about by digging into the specific types of results presented on the SERP. Further more, it may help us identify query ambiguity. By reviewing the content types of vertical



Fig. 3. Search Result Page(Query: Big Bang) with Results from Vertical Search Engines

results on SERPs, we extract 13 features by counting if a certain type of vertical result appears on the SERP, for instance, video, image, music, drama, news and etc. Each of them may describe a particular aspect of the queries, for example, if the there are direction results on the SERP, that may suggest that the query contains a name of a location.

3.3 Session Features

Query2vec Query session context may contain some information about the query ambiguity. By reviewing the session corpus, we intuitively find that for a clear query, the queries in session context usually focus on a specific topic while ambiguous queries are often surrounded by queries which covers many subtopics. For example, take the queries in Table 1 for example, the query "wind" is ambiguous, its preceding queries and following queries cover several domains or topics: Operating System (windows, wind7 download), Music (bboxwind, nobody knows, Dad I'm back), Weather (windy, wind speed) and Wind Information Co., Ltd (wind information, wind dataset, wind stock expert), which is a famous integrated service provider of financial data, information and software. Another query "high-resolution map of China" is clear and its context queries are mainly focused on the map of China, with differences in the details.

Inspired by recent distributional word representation approach [5], we attempt to present every query as a vector and extract features from both the original query and its "neighbors" vectors. We call it "query2vec" to shadow, as an application of "word2vec".

We further assume that context queries of a specific query, which carry vague intent, may scatter in the query vector space. If we treat a user *search session* as a *document*, *queries* in the sessions are just like *words* in the documents. We can learn a vector presentation for every query by training on large scale of user sessions. Formally, we can describe this the training process as the following

Query	Ambiguity	Identification	Based	on User	Behavior	Information
u ./						

8

Query: wind	(Ambiguous)	Query: high-resolution map of China (Clear)			
Pre. Queries(Freq)	Fol. Queries(Freq)	Pre. Queries(Freq)	Fol. Queries(Freq)		
bbox wind (2389)	windows (2432)	China $map(4362)$	China map (4783)		
wind info. (1897)	thunder(1701)	entire China $map(1541)$	entire China $map(1857)$		
windows (1765)	wind info. (1623)	China map high-reso (273)	pic of China $map(365)$		
wind dataset (1282)	wind dataset (1452)	China map $search(210)$	map(286)		
nobodyknows(453)	wind speed(432)	World $map(110)$	world $map(215)$		
Dad Im back(323)	win7 download (321)	entire world $map(106)$	Chinese province $map(210)$		
win(301)	win(286)	Chinese China $map(95)$	search on China $map(187)$		
telephone(211)	wind stock $expert(178)$	China map $copy(85)$	China map download (158)		
windy (153)	bboxwind(21)	pic of China map(81)	China map high-reso.(153)		

Table 1. Query Context of Query "wind" and "high-resolution map of China"

First, we collect a large scale of user sessions as a document $D = \{s_1, s_2, s_3...\}$, where s_i denotes the user sessions, each session contains a sequence of individual queries. Similar to [5], skip-gram model is adopted on this problem, the training objective is to learn query vector representations that are good at predicting the nearby queries. We chose a computationally efficient approximation of the full softmax, the hierarchical softmax, which will improve both the quality of the vectors and the training speed. One of applications with the vectors is to calculate similarities between any two items. For words in documents, the similarities means the possibility that the two words have similar meanings or share common properties, or propose the same connotation. For queries in sessions, the similarities between queries usually indicates the similarity of user intents, which the two queries reflect.

Features Extraction As stated before, each query can be presented as a vector with *query2vec* algorithm. For query q_i , we can pick out its top-k frequently occurred preceding and following queries separately as $P(\vec{q}_i, k)$ and $F(\vec{q}_i, k)$. each query can be presented by its vector, We can get two vector sets P^V and F^V by replacing queries by vectors,

$$P^{V}(\overrightarrow{v_{i}},k) = \{\overrightarrow{v_{i,1}}, \overrightarrow{v_{i,2}}, \overrightarrow{v_{i,3}}, \dots, \overrightarrow{v_{i,k}}\} \quad F^{V}(\overrightarrow{v_{j}},k) = \{\overrightarrow{v_{j,1}}, \overrightarrow{v_{j,2}}, \overrightarrow{v_{j,3}}, \dots, \overrightarrow{v_{j,k}}\} \quad (6)$$

where the queries is sorted by times of co-occurance in descending order.

We calculate *Diameter*, *Mean* and *Standard Deviation* (SD) on similarities between queries from both $P^V(\overrightarrow{v_i}, k)$ and $F^V(\overrightarrow{v_i}, k)$ to quantify the distribution of Preceding and Following queries. Click entropy is also chosen to describe the distribution of the two sets. The vector similarity is measured by cosine similarity. We utilize the open source toolkit $word2vec^1$ to train the query representations. The length of top co-occurred preceding and following queries is a parameter can be tuned.

¹ code.google.com/p/word2vec/

4 Experiments

4.1 Experiment Setup

With the help of a famous Chinese commercial search engine, we collect query logs from April 1, 2012 to April 30, 2012. We extract the information of the query, clicked URL, timestamp and anonymized user-id. Private information is removed as much as possible to avoid the leak of privacy information. The click-through log contains 2,625,029 unique queries, 4,699,150 unique URLs, 72,106,874 query-URL pairs, and 717,916,107 individual clicks.

Search sessions are divided according to the time interval between two queries issued by a user. We separate two consecutive queries into two sessions if the interval exceeds 30 minutes. After removing all the single-query sessions, there are 160,526,561 sessions left on which the "query2vec" algorithm is conducted.

To construct a training set, we randomly sample 1,000 queries from query logs. After removing the pornographic ones, there are 810 queries left. With the help of 2 professional assessors, we label each query as clear or not-clear according to the following labeling criteria:

CLEAR: the query reflects a clear user intent.

NOT CLEAR: the query may have different interpretations (Ambiguous) or covers several subtopics (Broad) of an entity.

After the annotators double-checked the annotations, the query-set contains 485 CLEAR queries and 325 NOT CLEAR queries in total. The κ value of annotation between three assessors is 0.756, which denotes the accessors reach agreements on most of the instances.

4.2 Classification Performance

By adopting the proposed query enrichment and feature extraction methods, several groups of features are extracted to describe the corresponding queries including: Basic Features (BF), User Click Features (UCF), Session Features (SF) and Search Result Hints (SRH). We train a SVM classifier to combine these features. Intuitively, we choose Precision, Recall and F1-Score for evaluation purposes just as other query ambiguity identification works[3, 4]. All the evaluation results are obtained with 5-folded cross validation. The length of top co-occurred preceding and following queries, parameters of SVM(cost and γ) are tuned to gain its best performance for comparison.

From Table 2, we can see that the *Basic Features* plays an important role in query ambiguity identification. Feature combinations 1 to 4 show the performance of classifiers with only one group of features. Since Basic Features get the best performance, we focus on the improvement brought by User Click Features, Session Features and Search Result Hints with the existence of Basic Features. With the number of involved feature groups increases, the classification performance also improves. Feature combinations No. 7 and No. 11 illustrates that Session Features help improve the performance (compared with Feature combinations No.1 and No. 9 respectively), which show the effectiveness of the

#	Feature	Precision	Recall	F_1 Score	F_1 Score Improvement Compared with Baseline
1	BF	0.756	0.747	0.748	-2.09%
2	UCF	0.624	0.619	0.568	-25.65%
3	SF	0.710	0.711	0.710	-7.07%
4	SRH	0.715	0.702	0.682	-10.73%
5	BF+UCF	0.789	0.789	0.789	3.27%
6	BF+SF	0.766	0.766	0.766	0.26%
7	BF+SRH	0.767	0.767	0.764	0.00%
8	BF+UCF+SRH	0.813	0.813	0.811	6.15%
9	BF+UCF+SF	0.798	0.798	0.797	4.32%
10	BF+SF+SRH	0.788	0.788	0.785	2.75%
11	BF+UCF+SF+SRH	0.821	0.820	0.819	7.20%
12	Hafernik et al. (Baseline)	0.769	0.763	0.764	N/A

10 Query Ambiguity Identification Based on User Behavior Information

Table 2. Performance Comparison with Different Features Combinations

proposed 'query2vec" method. **BF+UCF+SF+SRH** outperforms other feature combinations and archives Precision 0.821, Recall 0.820 and F-Score 0.819. Comparing with other combinations, we find that Basic Features is the most helpful for the query ambiguity problem and Search Result Hints, User Click Features and Session Features, also contribute to the classification performance.

The taxonomy of Narrow/General queries proposed by Hafernik et al. is quite similar with our Clear/Not-Clear from the ambiguity identification's perspective. Therefore, we also implemented their classification method as a baseline to compare with. As shown in Table 2, several of our feature combinations (including Combinations No. 5, 6, 8, 9, 10 and 11) outperforms this baseline method.

The classifier sometimes makes mistakes when predicting ambiguous queries. For example, the query '5566', it is the name of a famous boy band which was very active ten years ago, it is also the name of a well-known Chinese navigational website . However, most users select the results for navigation website, making the user behavior features (UCF, SF) bring negative influence to the classifier.

We divide the queries into two group: low&medium frequency group(0-1000 clicks) and high frequency group(1000 + clicks) and compare algorithm performance with all features on different queries groups.

Groups	Precision	Recall	F_1 Score
low & medium frequency queries	0.855	0.860	0.857
high frequency queries	0.806	0.799	0.802
Table 3. Performance Compariso	on on Differ	ent Que	ry Groups

Results from Table 3 show that our algorithm is of most help for low&medium frequency queries. It would be promising because the vast majority of queries processed by search engines is made up by low and medium frequency queries.

4.3 Comparison of Feature Contributions

Experiment is also conducted to evaluate the contributions of individual features. First, we train a model with **BF+UCF+SF+SRH** features, and the parameters are tuned to gain its optimal performance.

After that, we remove each feature from the above model with all features and calculate the performance loss caused by the feature removal. We use accuracy (the proportion of both true positive and true negative examples) to evaluate the differences with the original model. We can see that a larger performance loss represents a larger contribution of the corresponding feature to the classification framework. According to the results shown in Figure 4, *Query Length*



Fig. 4. Feature Contribution Comparison

contributes the most to this task. This effect can be explained by the fact that most ambiguous queries are relatively short, which accords with the conclusions given by [4, 11].

5 Conclusion and Future Work

Nowadays, a typical short query submitted to a search engine usually cannot offer enough information for the ranking algorithm to give a high quality result list. Therefore, the best a search engine can do is to provide a diversified or personalized search results. In this work, to identify query ambiguity, we propose a 3-level framework which takes user behavior features collected from click-through logs into consideration. Especially, besides the features collected from queries, we focus on how to tell the differences between clear and ambiguous queries via features extracted from multi-query sessions. Inspired by recent progresses in word representation, we propose a query representation approach named "query2vec" which constructs representations from the distributions of contexts in query logs. Experiments based on large scale commercial search engine logs show effectiveness of the proposed framework as well as the corresponding representation approach. In the future, we plan to adopt the ambiguity identification algorithm in practical search environment to guide the diversified or personalized ranking process. 12 Query Ambiguity Identification Based on User Behavior Information

References

- 1. Clarke, Charles LA, Maheedhar Kolla, and Olga Vechtomova. An effectiveness measure for ambiguous and underspecified queries. Springer, 2009.
- Speretta, Mirco, and Susan Gauch. "Personalized search based on user search histories." Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on. IEEE, 2005.
- Song, Ruihua, et al. "Identification of ambiguous queries in web search." Information Processing & Management 45.2 (2009): 216-229.
- 4. Song, Ruihua, et al. "Learning query ambiguity models by using search logs." Journal of Computer Science and Technology 25.4 (2010): 728-738.
- Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in Neural Information Processing Systems. 2013.
- Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
- Broder, Andrei. "A taxonomy of web search." SIGIR forum. Vol. 36. No. 2. ACM, 2002.
- 8. Rose, Daniel E., and Danny Levinson. "Understanding user goals in web search." Proceedings of the 13th international conference on World Wide Web. ACM, 2004.
- Zhai, Cheng Xiang, William W. Cohen, and John Lafferty. "Beyond independent relevance: methods and evaluation metrics for subtopic retrieval." in proceedings of SIGIR 2013. ACM, 2003.
- Chirita, Paul Alexandru, et al. "Using ODP metadata to personalize search." Proceedings of the 28th annual international ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2005.
- Hafernik, Carolyn Theresa, and Bernard J. Jansen. "Understanding the specificity of web search queries." CHI'13 Extended Abstracts on Human Factors in Computing Systems. ACM, 2013.
- Li, Ying, Zijian Zheng, and Honghua Kathy Dai. "KDD CUP-2005 report: Facing a great challenge." ACM SIGKDD Explorations Newsletter 7.2 (2005): 91-99.
- Shen, Dou, et al. "Building bridges for web query classification." in proceedings of SIGIR 2006. ACM, 2006.
- Beitzel, Steven M., et al. "Varying approaches to topical web query classification." Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007.
- 15. Westerveld, Thijs, Wessel Kraaij, and Djoerd Hiemstra. "Retrieving web pages using content, links, urls and anchors." (2002): 663-672.
- 16. Brenes, David J., and Daniel Gayo-Avello. "Automatic detection of navigational queries according to Behavioural Characteristics." LWA. 2008.
- Agichtein, Eugene, and Zijian Zheng. "Identifying best bet web search results by mining past user behavior." Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006.
- Liu, Yiqun, et al. "Automatic query type identification based on click through information." Information Retrieval Technology. Springer, 2006. 593-600.
- 19. Wang, Yu, and Eugene Agichtein. "Query ambiguity revisited: clickthrough measures for distinguishing informational and ambiguous queries." Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. ACL, 2010.
- 20. Lee, Uichin, Zhenyu Liu, and Junghoo Cho. "Automatic identification of user goals in web search." Proceedings of the 14th international conference on World Wide Web. ACM, 2005.