Context-Aware Query Classification

Huanhuan Cao[1]* Derek Hao Hu[2] Dou Shen[3] Daxin Jiang[4]
Jian-Tao Sun[4] Enhong Chen[1] Qiang Yang[2]
[1]University of Science and Technology of China [2]Hong Kong University of Science and Technology
[3]Microsoft Corporation [4]Microsoft Research Asia
[1]{caohuan, cheneh}@ustc.edu.cn [2]{derekhh, qyang}@cse.ust.hk [3,4]{doushen, djiang, jtsun}@microsoft.com

ABSTRACT

Understanding users' search intent expressed through their search queries is crucial to Web search and online advertisement. Web query classification (QC) has been widely studied for this purpose. Most previous QC algorithms classify individual queries without considering their context information. However, as exemplified by the well-known example on query "jaguar", many Web queries are short and ambiguous, whose real meanings are uncertain without the context information. In this paper, we incorporate context information into the problem of query classification by using conditional random field (CRF) models. In our approach, we use neighboring queries and their corresponding clicked URLs (Web pages) in search sessions as the context information. We perform extensive experiments on real world search logs and validate the effectiveness and efficiency of our approach. We show that we can improve the F_1 score by 52% as compared to other state-of-the-art baselines.

Categories and Subject Descriptors

H.3.m [**Information Storage and Retrieval**]: Miscellaneous; I.5.2 [**Pattern Recognition**]: Design Methodology-*Classifier design and evaluation*

General Terms

Algorithms, Experimentation

Keywords

Search context, Query classification

1. INTRODUCTION

Search engines have become one of the most popular tools for Web users to find their desired information. As a result, understanding the search intent behind the queries issued by Web users has become an important research problem. *Query classification* (or *query categorization*), denoted as QC, has been studied for this purpose by classifying user queries into a ranked list of predefined target categories. Such category information can be used to trigger the most appropriate vertical searches corresponding to a query, improve Web page ranking [18], and help find the relevant online advertisements.

Query classification is dramatically different from traditional text classification because of two issues. First, Web queries are usually very short. As reported in [5], most queries contain only 2-3 terms. Second, many queries are ambiguous [11], and it is common that a query belongs to multiple categories. For example, [27] manually labels 800 randomly sampled queries from the public data set from ACM KDD Cup'05[1], and 682 queries have multiple category labels.

To address the above challenges, a variety of query classification approaches have been proposed in the literature. In general, these approaches can be divided into three categories. The first category tries to augment the queries with extra data, including the search results returned for a certain query, the information from an existing corpus, or an intermediate taxonomy [8, 27]. The second category leverages unlabeled data to help improve the accuracy of supervised learning [5, 6]. Finally, the third category of approaches expands the training data by automatically labeling some queries in some click-through data via a self-training-like approach [21]. Although the existing methods may be successful in some cases, most of them are not context-aware; that is, they treat each query individually without considering the user behavior history.

A MOTIVATING EXAMPLE. Suppose that a user issues a query "*Michael Jordan*". It is not clear whether the user is interested in the famous basketball player or the machine learning researcher at UC Berkeley. Without understanding the user's search intent, many existing methods may classify the query into both categories "Sports" and "Computer Science". However, if we find that the user has issued a query "*NBA*" before "*Michael Jordan*", it is likely that the user is interested in the category of "Sports". Conversely, if the user issues some queries related to machine learning before the query "*Michael Jordan*", it may suggest the user is interested in the topics related to "Computer Science".

*The work was done when Huanhuan Cao and Derek Hao Hu were interns at Microsoft Research Asia.

[1]ACM KDD Cup'05 is an open contest conducted in conjunction with the ACM KDD'05 conference, which gives a QC task on 800,000 randomly selected Web queries.

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

Intuitively, using search context information, such as the adjacent queries in the same session as well as the clicked URLs of these queries, can help better understand users' search intent and thus improve the classification accuracy. As shown in previous studies (e.g., [9, 10, 13]), adjacent queries raised by the same user are usually semantically related. Moreover, compared with search queries, which are often short and ambiguous, the URLs that are selectively clicked by a user after issuing the queries may better reveal the search intent of the user.

As a first attempt to leverage context information in query classification, in this paper, we intend to answer the following questions: 1) How do we model context information effectively and incorporate it into the problem of query classification? 2) How much improvement can we achieve by using context information in query classification? 3) Would incorporating context information add too much computational burden and would it be possible to extend the idea for real world commercial search engines?

To answer these questions, we propose to use the *Conditional Random Field* model (*CRF* for short) [19] to help incorporate the search context information. We have several motivations for using this model. First, CRF is a sequential learning model which is particularly suitable for capturing the context information of queries. Second, the CRF model does not need any prior knowledge for the type of conditional distribution. Finally, compared with Hidden Markov Models, the CRF model is more flexible to incorporate richer features, such as the knowledge of an external Web directory.

In this paper, we show how CRF can be used for modeling the context information for query classification. We conduct extensive experiments on real world search logs to empirically evaluate our proposed model. Our experiments show that the CRF approach improves the F_1 score by as much as 52% as compared to the state-of-the-art baselines. Moreover, after the CRF model is trained offline, the online inference stage is very fast (e.g., less than 0.1 millisecond in our experiments), which makes our approach feasible to use in real world search engine systems.

The rest of this paper is organized as follows. In Section 3, we describe the problem of context-aware query classification. We then briefly introduce the CRF model in Section 4 and present the features in CRF in Section 5. The experimental results are reported in Section 6 and the related work is discussed in Section 2. Finally, we conclude our paper and point out some future research directions in Section 7.

2. RELATED WORK

Given a query and a predefined taxonomy, the objective of query classification (QC) is to classify the query into a ranked list of categories which are leaf nodes of the taxonomy. Previous studies on QC can be classified into two categories depending on the types of taxonomy.

In the first category, the taxonomy is defined by considering the Web query types. In [7], Broder gave the first taxonomy of Web query types such as Navigational Queries, Informational Queries and Transactional Queries. Rose et al. [24] introduced a more complex taxonomy of Web query types based on a popular taxonomy proposed by Broder. However, both of their works do not deal with classification based on a taxonomy of categories. Lee et al. [20] studied the classification problem and introduced an approach to classify Web queries into either Informational Queries or Navigational Queries. But their approach does not consider Transactional Queries. Recently, the problem of detecting commercial intent (OCI) attracts some researchers' interest [12]. This problem is also a QC problem by considering Web queries types.

For the second category, a taxonomy is defined by considering the topics of queries. Early work of query topic classification was done by manually classifying Web queries for query analysis, especially on the query topic distribution [4]. Since it is expensive for manually classifying Web queries, it is an interesting problem to design a automatic approach for classifying Web queries with a taxonomy of topics. Early works on this problem only considered the local information of queries, i.e., the terms of queries [5, 6]. However, as mentioned by [8], queries are usually short and the internal information is very limited. Recently, some works proposed to leverage the external Web knowledge to enrich the queries, such as extracting information from the top related search results of the query from a search engine [8] or taking advantage of a Web directory [27]. Given the fast growth of non-English web, some researchers studied the problem of cross-language query classification [23]. The approach proposed in this paper does not consider different languages.

In recent years, some researchers realized the importance of search context. In [16] and [9], two context-aware approaches to query suggestion were proposed. Cao et al. [10] proposed a general context-aware model for query suggestion and ranking. These works confirmed that search contexts are effective for disambiguating Web queries and can help improve the quality of multiple search services. However, to the best of our knowledge, none of existing works on query classification considered the search context. In our approach, click information is considered as an important part of context. Though many existing works such as [9, 29] studied how to use click-information to enrich query's semantic feature, none of them proposed an approach for leveraging click information for QC.

Proposed by Lafferty et al in [19], CRFs have been widely used in various domains such as named-entity recognition (NER) [22], identifying protein names in biology abstracts [26] and Web query refinement [14]. Some researchers also studied some variants of the basic Linear chain-CRF such as Skip-chain CRF [28]. In this paper, we use the basic Linear-chain CRF to model the query context.

3. PROBLEM STATEMENT

In this section, we introduce several notations and then give a description of *context-aware query classification* based on the definition of traditional query classification problem in [27].

Definition 1. (Search context and contextual queries). A user search session $\mathbf{o}$ is a series of observations $o_1 \dots o_T$, where each observation $o_t (1 \leq t \leq T)$ consists of a query q_t and a set of URLs U_t clicked by the user for q_t. For any query $q_k (1 < k \leq T)$, the observations $o_1 \dots o_{k-1}$ is the *search context* of q_k. In particular, the series of queries $q_1 \dots q_{k-1}$ are called the *contextual queries* of q_k. ■

Various methods have been proposed for session segmentation in the literature [17]. In this paper, we adopt a simple yet effective rule to divide sessions [9]; that is, two user

queries are divided into different sessions if the interval between their issued times is longer than 30 minutes. This rule has been widely used in previous works, e.g., [9, 16], and proved effective. Table 1 shows some examples of real user sessions. The symbol "↓" indicates the user clicks a URL. Note that the query "*gmc*" is an ambiguous query that refers to the "GMC cars" in session S_1, but refers to the "General Medical Council of Britain" in session S_2.

SID	User session
S_1	ford ⟹ toyota ⟹ gmc ↓ ↓ ↓ www.ford.co.uk www.toyota.com www.gmc.com
S_2	registered nurse ⟹ gmc ↓ ↓ www.mayo.edu/mshs www.gmc-uk.org
S_3	ancient Rome ⟹ gladiator ↓ ↓ www.historyforkids.org en.wikipedia.org/wiki/Gladiator

Table 1: Examples of real user sessions.

DEFINITION 2. (TAXONOMY). A *taxonomy* Υ is a tree of categories where each node corresponds to a predefined category. The semantic meaning of each category is defined by the labels along the path from the root to the corresponding node. ∎

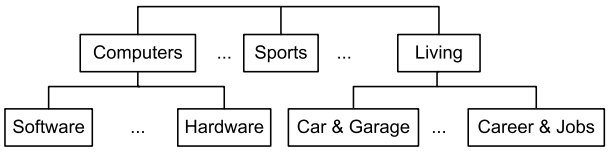


Figure 1: An example of taxonomy.

Figure 1 shows a part of the taxonomy of ACM KDD Cup'05. Sometimes we need to map the categories of a taxonomy Υ to the categories of another taxonomy $\widehat{\Upsilon}$ for reusing the category labels of Υ. In this paper, we adopt an effective mapping method introduced by Shen et al. [27] as a part of our approach.

DEFINITION 3. (LEVEL-N CATEGORY, ANCESTOR CATEGORY, AND SIBLING CATEGORY). Given a category c in a taxonomy Υ, c is called a *level-n category* if the node of c is located at n-th level of Υ. A category c^* is a *level-m ancestor category* of c, denoted by α_c^m $(m < n)$, if c^* is a level-m category and c^* corresponds to an ancestor node of c in Υ. A category $c^{\#}$ is an *level-m sibling category* of c, denoted by β_c^m $(m < n)$, if $c^{\#}$ is at the same level with c and $c^{\#}$ shares a common ancestor category α_c^m with c. ∎

For instance, in Figure 1, given a level-2 category c "Living \Car& Garage", α_c^1 is the level-1 ancestor category "Living" and "Living\Career& Jobs" is a level-1 sibling category of c.

Problem Statement (CONTEXT-AWARE QUERY CLASSIFICATION). Given a target taxonomy Υ, a user-specified parameter K, and a user query q_T, *context-aware query classification* incorporates the search context of q_T to classify q_T into a ranked list of K categories c_{T1}, c_{T2},..., c_{TK}, among N_c leaf categories $\{c_1, c_2, ..., c_{N_c}\}$ of Υ.

4. MODELING SEARCH CONTEXT BY CRF

The *Conditional Random Field* (CRF) model is a discriminative graphical model, which focuses on modeling the conditional distribution of unobserved state sequences given an observation sequence [19]. The strength of processing sequential data and incorporating rich features makes CRF model particularly suitable for context-aware query classification.

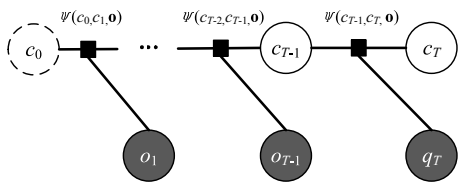


Figure 2: Modeling search context by a Linear-chain CRF.

As shown in Figure 2, in our problem, a Linear-chain CRF defines the conditional probability of a category label sequence $\mathbf{c} = c_1...c_{T-1}c_T$ given an observation sequence $\mathbf{o} = o_1...o_{T-1}q_T$ as:

$$p(\mathbf{c}|\mathbf{o}) = \frac{1}{Z(\mathbf{o})} \prod_{t=1}^{T} \psi(c_{t-1}, c_t, \mathbf{o}), \tag{1}$$

where $Z(\mathbf{o}) = \sum_{\mathbf{c}} \prod_{t=1}^{T} \psi(c_{t-1}, c_t, \mathbf{o})$ is a normalization factor and c_0 is an empty category label which is added for simplicity of defining the model. Potential functions ψ describe the Linear-chain transitions, and are defined as:

$$\psi\left(c_{t-1}, c_t, \mathbf{o}\right) = \exp\left(\sum_k \lambda_k f_k\left(c_{t-1}, c_t, \mathbf{o}\right)\right), \tag{2}$$

where f_k is a feature function and λ_k is the weight of f_k. Given training data $\mathcal{D} = \{\mathbf{o}^{(n)}, \mathbf{c}^{(n)}\}_{n=1}^{N}$, the objective of training a Linear-chain CRF is to find a set of parameters $\Lambda = \{\lambda_k\}$ that maximize the conditional log-likelihood:

$$L(\Lambda) = \sum_{n=1}^{N} \log p(\mathbf{c}^{(n)}|\mathbf{o}^{(n)}). \tag{3}$$

Once the parameters Λ have been learned using a training data set, we can infer the category label c_T^* for the test query q_T as $c_T^* = \arg\max_{\mathbf{c_T}} p(c_T|\ \mathbf{o}, \Lambda)$.

5. FEATURES OF THE CRF MODEL

When we use the CRF to model a search context, one of the most important parts is to choose the effective feature functions. In this section, we introduce the features used for building a CRF model of the search context for QC. In general, the features can be divided into two categories. The features that do not rely on the context information are called *local features*, and those that are dependent on context information are called *contextual features*.

5.1 Local features

To leverage the local information of individual queries, we consider three types of features that associate queries with the corresponding category label, namely, query terms, pseudo feedback, and implicit feedback.

5.1.1 Query terms

Given a query q_t $(1 \leq t \leq T)$ and its category label c_t, the elementary features that reflect the association between q_t and c_t are the terms of q_t. Suppose q_t consists of a set of terms $\{t_{q_t}\}$, each t_{q_t} can be considered as a feature to support the category label c_t. The weights of these features can be learned in the training process of the CRF model.

The problem of this type of features is that query terms are usually sparse. Consequently, the available training data are usually with limited size and could not cover a sufficient set of query terms that are useful for reflecting the association between queries and category labels. Therefore, given a new query whose partial, or all terms do not occur in the training data, this kind of features will not work.

The above problem is difficult to solve because it is hard to label a large number of sessions with a complex taxonomy for a sufficiently large set of terms for all categories. For this reason, we also consider some other features that represent the association between queries and category labels by leveraging some external Web knowledge.

5.1.2 Pseudo feedback

This type of features exploits the top M results returned by an external Web directory. Given a query q_t $(1 \leq t \leq T)$ and its category label c_t, we first submit q_t to an external Web directory, such as the Google Directory [2] or Yahoo Directory [1], and get the top M search results. In the second step, for each of the top-M results, we follow the method in [27] and map its category label from a category in the Web directory's taxonomy to a category in the target taxonomy. Finally, we calculate a *general label confidence score*:

$$GConf(c_t, q_t) = \frac{M_{c_t,q_t}}{M},$$

where M_{c_t,q_t} means the number of returned related search results of q_t whose category labels are c_t after mapping. Intuitively, the $GConf$ score reflects the confidence that q_t is labeled as c_t gained from pseudo feedback; the larger the score, the higher the confidence.

5.1.3 Implicit feedback

The third type of local features considers the click information as the implicit feedback from users. Similar to the type of features from pseudo feedback, we also exploit an external Web directory. However, we use the clicked URLs by users instead of the top-M results returned by the Web directory to enrich queries. To be more specific, given a query q_t $(1 \leq t \leq T-1)$, let the set of clicked URLs of q_t be $U_t = \{u_t\}$, the *click-based label confidence score* of c_t given q_t is defined as:

$$CConf(c_t, q_t, U_t) = \frac{\sum_{u_t} CConf(c_t, u_t)}{|U_t|},$$

where $CConf(c_t, u_t)$ means the confidence that c_t is the most appropriate category label of u_t.

We calculate $CConf(c_t, u_t)$ in three steps. The first two steps are similar to those in calculating the general label confidence score. That is, we first submit q_t to a Web directory and then map the category of each top-M result to a corresponding category in the target taxonomy. After these two steps, we obtain a document collection for each possible category of q_t in the target taxonomy, which will be used to calculate $CConf(c_t, u_t)$. In the third step, we build a *Vector Space Model* (VSM) [25] for each category from its document collection and make the cosine similarity between the term vector of c_t and the term vector of u_t as $CConf(c_t, u_t)$. The snippets of the web pages are used for generating the term vectors.

It is a special case that the top-M search results returned by the Web directory contain the clicked URL u_t. In this case, u_t is associated with a Web directory label $\widetilde{c}_t$. Denoting the mapped category label of $\widetilde{c}_t$ as $\widehat{c}_t$, we define $CConf(\widehat{c}_t, u_t) = 1$ and $\forall_{c \neq \widehat{c}_t} CConf(c, u_t) = 0$.

Note that the $CConf$ score is only applicable when the click information of q_t is available. If a user does not click on any URL for q_t, or q_t is the current query to be classified, this score cannot be calculated.

5.2 Contextual features

To use the context information, we consider some features that can reflect the association between adjacent category labels.

5.2.1 Direct association between adjacent labels

Occurrence of a pair of adjacent labels $\langle c_{t-1}, c_t \rangle$ $(1 < t \leq T)$ is an obvious feature of the association between adjacent labels, where c_{t-1} and c_t are leaf categories in the target taxonomy. The higher the weight $\langle c_{t-1}, c_t \rangle$, the larger the probability c_{t-1} transits into c_t. The weights of these features are learned from the training data during the training process of the CRF model.

5.2.2 Taxonomy-based association between adjacent labels

Limited by the size of the training data, some transition between categories may not occur in the training data. Moreover, the number of observed transitions may not reflect the distribution in real world applications. Consequently, the CRF model may not be able to capture the direct association between categories properly.

To reduce the bias of training data, besides considering the feature of direct association between adjacent labels, we also consider the structure of the taxonomy. Intuitively, the association between two sibling categories is stronger than that of two non-sibling categories. For example, the category "Computer\Software" is more relevant to "Computer\Hardware" than to "Live\Career& Jobs". Please refer to Definition 3 for the formal definition of sibling categories.

To be more specific, given a pair of adjacent labels $\langle c_{t-1}, c_t \rangle$, where c_{t-1} and c_t are both leaf categories at level n, we consider $n-1$ features of taxonomy-based association between c_{t-1} and c_t as $\{\langle \alpha^i_{c_{t-1}}, \alpha^i_{c_t} \rangle\}$ $(1 \leq i \leq n-1)$. The weights of these features are learned from the training data. This idea is similar to smoothing, where, if there are no training data for the feature when $\langle c_{t-1}, c_t \rangle$ occurs, there may still be some training data for the features at higher-level transitions $\langle \alpha^i_{c_{t-1}}, \alpha^i_{c_t} \rangle$ in the training data. Let $\beta^i_{c_{t-1}}$ and $\beta^i_{c_t}$ be the level-i siblings of c_{t-1} and c_t, respectively. It is easy to see that $\langle \alpha^n_{c_{t-1}}, \alpha^n_{c_t} \rangle$ occurs if $\exists \beta^n_{c_{t-1}}, \beta^n_{c_t}$ such that $\langle \beta^n_{c_{t-1}}, \beta^n_{c_t} \rangle$ occurs.

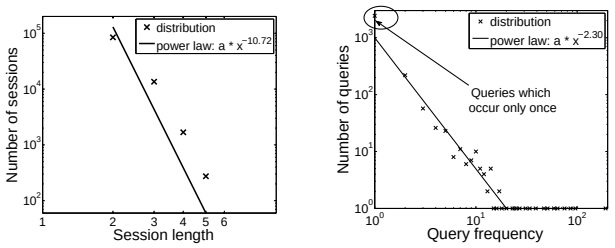
6. EXPERIMENTS

In this section, we validate our proposed methods through a systematic empirical comparisons with two baselines over a real data set.

6.1 Experimental Set Up and Data Sets

We use the target taxonomy of ACM KDD Cup'05 as our target taxonomy, which is widely used in the literature for QC. This taxonomy is a two-level taxonomy and has seven level-1 categories and 67 level-2 categories.

We randomly extract 10,000 sessions from one day's search log of a major commercial search engine under the session segmentation rule mentioned in Section 3. In this paper, all the extracted sessions contain at least two queries so that we can exploit the impact of contextual information for query classification. The proportion of sessions with more than one query is usually not small. As shown in [15], about 55% sessions have more than one query in a search log of the Excite search engine. Our search log also shows that there are more than 45% such sessions among all sessions. It implies that our approach can help in many cases. Moreover, the queries in single query sessions are mostly "easy queries" that have clear meanings and are easy to be classified. In the extracted sessions, there are 23,091 unique queries and 32,410 unique clicked URLs in total.



(a) (b)

Figure 3: Distributions of (a) session lengths and (b) query frequencies of the training data.

Figure 3 (a) and Figure 3 (b) show the session length distribution and the query frequency distribution of the data set, respectively. From these two figures we can see that in this data set, both the distribution of session lengths and the distribution of query frequencies roughly follow the power law. This phenomenon is consistent with some previous analysis on large scale search logs [9].

We invited three human labelers to label the queries of each session with the 67 level-2 category labels. For each query, a labeler gives a most appropriate category label by considering not only the query itself, but also the search context and the clicked URLs of the query. A query's final label is voted by the three labelers. Since each query is associated with context information (except for the beginning queries of sessions) and real user clicks which can help determine the meaning or intent of the query, the consistency among the labelers is quite high. For more than 90% queries, the three labelers give the same labels. This is very different from the general query classification problem [27].

Figure 4 shows the category distribution of the labeled queries. From this figure, we can see the category labels of the queries in our data set cover all seven level-1 categories.

6.2 Baselines

In this paper, we adopt two baselines to evaluate the performance of our approach:

Bridging classifier(BC): We implement the *bridging classifier* introduced by Shen et al in [27]. The idea of this approach is training a classifier on an intermediate taxonomy and then bridging the queries and the target taxonomy in the online step of QC. Experiments in [27] show this approach outperforms the wining approach in KDD Cup'05.

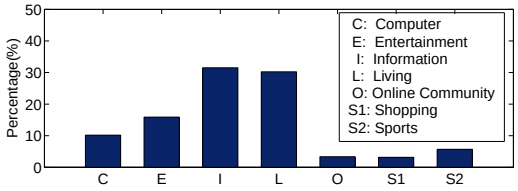


Figure 4: Distribution of different category labels in the training data.

Collaborating classifier(CC): Since there is no existing approach for query classification that takes into account the context information, we design a naive context-aware approach as the second baseline to further evaluate the modeling power of CRF in this problem. The idea of this approach is as follow: given a test query q_T and the previous query q_{T-1} in the same session, 1) firstly we use the bridging classifier to obtain all possible categories of q_T as $C_{q_T} = \{c_{q_T}\}$ with scores $Score(q_T, c_{q_T})$ and all possible categories of q_{T-1} as $C_{q_{T-1}} = \{c_{q_{T-1}}\}$ with scores $Score(q_T, c_{q_{T-1}})$; 2) After that, for each c_{q_T}, we let:

$$Score(c_{q_T}) = Score(q_T, c_{q_T}) + \sum_{c_{q_{T-1}}} Score(q_{T-1}, c_{q_{T-1}}) \times AConf(c_{q_{T-1}}, c_{q_T}),$$

where $AConf(c_{q_{T-1}}, c_{q_T})$ means the *association confidence* [3]of the adjacent label pair $\langle c_{q_{T-1}}, c_{q_T}\rangle$. The association confidence which is calculated as:

$$AConf(c_{q_{T-1}}, c_{q_T}) = \frac{freq(c_{q_{T-1}}, c_{q_T})}{\sum_c freq(c_{q_{T-1}}, c)},$$

where $freq(c1, c2)$ means the frequency of the adjacent label pair $\langle c1, c2\rangle$ in the training data. Finally the category label ranked list of C_T is generated by ranking $Score(c_{q_T})$.

6.3 Evaluation Metrics

Given a test session $q_1 q_2 ... q_T$, we take the last query q_T as the test query and take the queries $q_1 q_2 ... q_{T-1}$ and their corresponding clicked URL sets $U_1 U_2 ... U_{T-1}$ as the search context. In order to evaluate the performance of our approach and the two baselines on the task of query classification with search context, we use three metrics, namely, overall precision, overall recall and overall F_1 score. For a test query q_T with the true category label c_T, given the classification results $C_{T,K}$ where $C_{T,K}$ is a set of the top K predicted category labels from a tested approach, the precision(P) for q_T is represented as $\frac{\delta(c_T \in C_{T,K})}{|K|}$, where $\delta(*)$ is a boolean function of indicating whether $*$ is true(=1) or false(=0). The recall(R) for q_T is represented as $\delta(c_T \in C_{T,K})$ and the F_1

score for q_T is represented as $\frac{2\times P\times R}{P+R}$. The overall precision is calculated as $\frac{\sum_{n=1}^{N} P_n}{N}$, where N means the number of all test cases and P_n means the precision for the nth test query. The overall recall and overall F_1 score are both calculated in similar ways.

To reduce the uncertainty of splitting the data into training data and test data, we adopt a ten-fold cross validation as follow: 1) Firstly we randomly partition the labeled sessions into ten folds; 2) Then we take each of the ten folds as test data and the remaining nine folds as training data; 3) Finally, we report the average performance of the ten runs.

6.4 Overall Results and Analysis

In order to study the contribution of context information, we compare three CRF models with different features: CRF-B (*CRF with Basic features*[2]), CRF-B-C (*CRF with Basic features + Click-based label confidence*) and CRF-B-C-T (*CRF with Basic features + Click-based label confidence + Taxonomy-based association*), respectively. In our experiments, we choose Google Directory as our external Web directory for calculating general label confidence and click-based label confidence. We set M, i.e., the number of used search results of a Web directory, to be 10, which equals the number of search results in a search page.

In this section, we evaluate the overall precision, overall recall and overall F_1 score with different K for each tested approach. We set the maximum K to be 5.

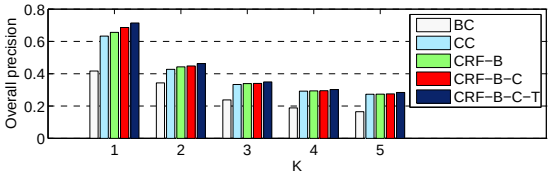


Figure 5: The average overall precision of CRF-B, CRF-B-C, CRF-B-C-T and two baselines with different K.

Figure 5 compares the average overall precision of CRF-B, CRF-B-C, CRF-B-C-T to the two baselines with different K values. From this figure we can see that all tested approaches' average overall precision numbers drop when we increase K. Compared with the non-context-aware baseline BC, the average overall precision of CRF-B, CRF-B-C and CRF-B-C-T is improved across different K by 50%, 52% and 57% , respectively. Compared with the naive context-aware baseline CC, average overall precision of CRF-B, CRF-B-C and CRF-B-C-T is also improved by 2%, 3% and 7%, respectively.

Similarly, Figure 6 compares the average overall recall of CRF-B, CRF-B-C, CRF-B-C-T and the two baselines with different K. From this figure we can see that all tested approaches' average overall recall values increase when we increase K. It is reasonable because the probability that the ground truth label is covered by the predicted results will increase with more predicted category labels. Compared with the non-context-aware baseline BC, the average overall recall of CRF-B, CRF-B-C and CRF-B-C-T is improved across

[2]*Basic features* mean Query terms, General label confidence and Direct association between adjacent labels

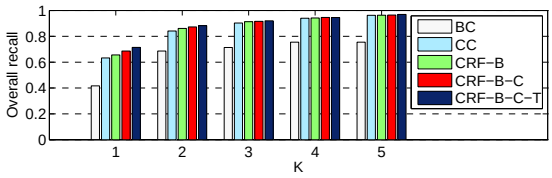


Figure 6: The average overall recall of CRF-B, CRF-B-C, CRF-B-C-T and two baselines with different K.

different K by 33%, 35% and 37% , respectively. Compared with the naive context-aware baseline CC, the average overall precision of CRF-B, CRF-B-C and CRF-B-C-T is also improved by 2%, 3% and 4%, respectively.

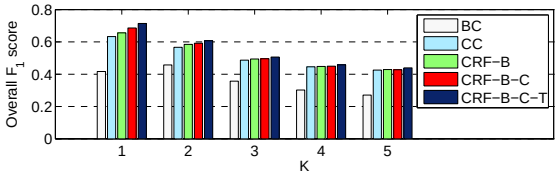


Figure 7: The average overall F_1 scores of CRF-B, CRF-B-C, CRF-B-C-T and two baselines with different K.

Figure 7 compares the average overall F_1 scores of CRF-B, CRF-B-C, CRF-B-C-T and the two baselines with different K. From this figure, we can see the CRF-B, CRF-B-C and CRF-B-C-T can improve the average F_1 scores by 46%, 48% and 52%, respectively, when compared to the non-context-aware baseline BC. Compared with the naive context-aware baseline CC, the average overall F_1 scores of CRF-B, CRF-B-C and CRF-B-C-T are also improved by 2%, 3% and 6%, respectively.

We conduct a series of paired T-tests of 0.95 confidence level which show that the improvements of our approaches on overall precision, overall recall and overall F_1 are all statistically significant. We also study the variances of overall precision, overall recall and overall F_1 scores of all tested approaches in the ten-fold cross validation. Table 2 shows the mean deviations of these values of each tested approach in the ten-fold cross validation with $K = 1$ and $K = 2$, respectively. Notice that when $K = 1$, the overall precision, overall recall and over all F_1 scores are same for each tested approach. From this table we can see that the variances of all three CRFs' performance are consistently smaller than the collaborating classifier. It implies that there is indeed a major advantage of using CRFs for extracting context information, as compared to the collaborating classifier based on a naive context-aware strategy.

We also compare the performance of our proposed approaches and the two baseline methods on user-session data with different lengths, where the shortest length is two. From the experiments, we find that the performance of all tested approaches on length-two sessions is a little better than sessions with more queries. This is because it is often the case that the shorter the sessions are, the more likely the queries are common queries that are easy to be classi-

K	Approach	Overall P	Overall R	Overall F_1
1	BC	6.77×10^{-3}	-	-
	CC	1.97×10^{-2}	-	-
	CRF-B	7.87×10^{-3}	-	-
	CRF-B-T	1.03×10^{-2}	-	-
	CRF-B-C-T	1.8×10^{-2}	-	-
2	BC	6.48×10^{-4}	1.30×10^{-3}	8.64×10^{-4}
	CC	1.58×10^{-2}	2.43×10^{-2}	1.94×10^{-2}
	CRF-B	2.31×10^{-3}	6.34×10^{-3}	3.47×10^{-3}
	CRF-B-C	5.57×10^{-3}	1.17×10^{-2}	7.47×10^{-3}
	CRF-B-C-T	6.81×10^{-3}	9.88×10^{-3}	8.20×10^{-3}

Table 2: Mean deviations of overall precision, overall recall and overall F_1 scores of each tested approach in the ten-fold cross validation.

fied. Moreover, for sessions with more than two queries, we compare the performance of CRFs by considering different lengths of search context. We find that considering longer search context does not significantly improve the performance as compared to considering only one previous query and its corresponding clicked URLs.

From the above experiments, we can come to the following conclusions: 1) Firstly, all three CRF models and collaborating classifier consistently outperform the bridging classifier on the task of query classification given search context, which implies the effectiveness of context information; 2) Secondly, all three CRF models consistently outperform the collaborating classifier, which is a naive context-aware baseline. It implies that it's an effective approach of modeling context information by CRFs; 3) Thirdly, CRF-B-C outperforms CRF-B, which shows that click information is a good source of context information for query classification; 4) Finally, CRF-B-C-T outperforms CRF-B-C, which indicates that the taxonomy-based association between adjacent labels is useful for the query classification problem with search context.

6.5 Case Study

In addition to the study on the overall performance of CRF-B, CRF-B-C, CRF-B-C-T and the two baselines, we also study the cases in which our approach outperforms the baselines.

Context info: travel guide → www.worldtravelguide.net	
Query: santa fe new mexico	
Snippet of the clicked URL: Santa Fe Travel Information and Travel Guide - USA - Lonely Planet	
Ground truth: Living\Travel & Vacation	
Category Labels	
Bridging classifier	Information\Local & Regional **Living\Travel & Vacation**
Collaborating classifier	**Living\Travel & Vacation** Information\Local & Regional
CRF-B-C-T	**Living\Travel & Vacation** Information\Local & Regional

Table 3: An example of query classification with a search context.

Table 3 shows an example of query classification with a search context. In this example, the test query is "*santa fe new mexico*". Without considering the context, this query may have multiple possible search intents. One possible intent is that the user wants to know some general information of the city of Santa Fe, such as the area, the population of this city, etc. In this case, the query should be classified into the "Information\Local & Regional" category. Another possible intent is that the user wants to go on a vocation in the city of Santa Fe and need some travel information about this city, such as hotels and tourist attractions. In this case, the query should be classified into the "Living\Travel & Vacation" category. However, given the context with the query "*travel guide*" in which the user visits a web site related to travel, the appropriate category of this query should be narrowed down to "Living\Travel & Vacation". From Table 3, we can see that both CRF-B-C-T and the collaborating classifier give the correct category label in the first position because they consider contextual information, while the bridging classifier's first label is not appropriate. This case exemplifies the effectiveness of considering context information.

Context info: FIFA → fifa08.ea.com	
Query: FIFA news	
Snippet of the clicked URL: FIFA 08 News, Videos	
Ground truth: Entertainment\Games & Toys	
Category Labels	
Bridging classifier	Sports\Soccer **Entertainment\Games & Toys**
Collaborating classifier	Sports\Soccer **Entertainment\Games & Toys**
CRF-B-C-T	**Entertainment\Games & Toys** Sports\Soccer

Table 4: Another example of query classification with a search context.

Table 4 shows another example of query classification given the search context. In this example, the test query is "*FIFA news*". Without considering the context, this query may have two possible meanings: news of the International Federation of Association Football, or news on a soccer video game named "*FIFA*". And the corresponding categories are "Sports\Soccer" and "Entertainment\Games & Toys", respectively. However, given the context that the user has issued a query "*FIFA*" and clicked a URL which is related to the video game "FIFA" , the appropriate category is most likely "Entertainment\Games & Toys". From Table 4 we can see that CRF-B-C-T gives the correct category label in the first position, while the collaborating classifier and bridging classifier's first labels are not appropriate. This case exemplifies that CRF-B-C-T leverage search context better than the collaborating classifier.

6.6 Efficiency of Our Approach

Our approach consists of an offline part and an online part. In the offline part, the time cost of our approach comes from the training cost for the CRF model. Figure 8 (a), (b) and (c) show the convergence curves of CRF-B, CRF-B-C and CRF-B-C-T, respectively. From these figures we can see the objective function value of CRF-B-C converges to a better optima as compared to CRF-B and the objective function value of CRF-B-C-T converges to a better optima point than CRF-B-C. This implies that considering click information and taxonomy-based association between adjacent category labels can help build a stronger CRF model. The training algorithms are implemented on an Intel Core2 2×2.0G, 4G

main memory machine. Each iteration of these algorithms takes about 300 milliseconds. Therefore, the time cost of training a CRF is acceptable as an off line process.

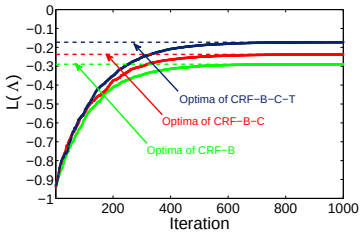


Figure 8: Objective function values per iteration of training CRF-B, CRF-B-C and CRF-B-C-T.

It is well known that Web users often have strict requirements on the response time of online applications. Thus, the efficiency of an online application is an important problem. In the online part, the time cost of our approach comes from calculating features and inference. In the stage of calculating features, the main cost comes from the process of calculating label confidence. This process can be very fast for a commercial search engine since most modern search engines have their own Web directories locally. Moreover, if we calculate these features offline in advance and store them in local servers, the process will be even faster. Besides, the stage of reference is very fast (less than 0.1 millisecond). This is because usually the length of search context is short and the number of possible categories for a query is small as well. For improving the efficiency of inference further, we can consider only one previous query and its corresponding clicked URLs as search context, since our experiments show that such context information is effective enough for improving the quality of QC significantly.

7. CONCLUSIONS AND FUTURE WORK

Web query classification is an important problem with wide applications. However, although many existing works have studied this problem, none of them considered the search context together with query classification. In this paper, we propose a novel approach for leveraging context information to classify queries by modeling search context though CRFs. Experiments on a real data set extracted from a commercial search engine log clearly show that our approach consistently outperforms a non-context-aware baseline and a naive context-aware baseline.

Our current approach cannot handle the first-query problem well, which is the problem of not being able to find a search context if the query is located at the beginning of a search session. However, if we can capture some events that occurred a little earlier at the beginning of the session, such as events of Web page browsing, we can solve the first query problem well. In our future research, we plan to study this problem in detail.

8. ACKNOWLEDGEMENT

Huanhuan Cao and Enhong Chen thank the support of MSRA Internet Services Theme. Qiang Yang and Derek Hao Hu thank the support of Microsoft Research project MRA07/08.EG01.

9. REFERENCES

[1] http://search.yahoo.com/dir.
[2] http://www.google.com/dirhp?hl=en.
[3] Angrawal, R., et al. Mining association rules between sets of items in large databases. In *SIGMOD'93*, pages 206–217, 1993.
[4] Beitzel, S., M., et al. Hourly analysis of a very large topically categorized web query log. In *SIGIR'04*, pages 321–328, 2004.
[5] Beitzel, S., M., et al. Automatic web query classification using labeled and unlabeled training data. In *SIGIR'05*, pages 581–582, 2005.
[6] Beitzel, S., M., et al. Improving automatic query classification via semi-supervised learning. In *ICDM'05*, pages 42–49, 2005.
[7] Broder, A., Z. A taxonomy of web search. In *SIGIR Forum*, pages 3–10, 2002.
[8] Broder, A., Z., et al. Robust classification of rare queries using web Knowledge. In *SIGIR'07*, pages 231–238, 2007.
[9] Cao, H., et al. Context-aware query suggestion by mining click-through and session data. In *KDD'08*, pages 875–883, 2008.
[10] Cao, H., et al. Towards context-aware search by learning a very large variable length hidden markov model from search logs. *To appear in WWW'09*, 2009.
[11] Cui, H., et al. Probabilistic query expansion using query logs. In *WWW'02*, pages 325–332, 2002.
[12] Dai,H.,K., et al. Detecting online commercial intention (oci). In *WWW'06*, pages 829–837, 2006.
[13] Fonseca, B.M., et al. Concept-based interactive query expansion. In *CIKM'05*, pages 696–703, 2005.
[14] Guo, J.,F., et al. A unified and discriminative model for query refinement. In *SIGIR'08*, pages 379–386, 2008.
[15] He, D., et al. Detecting session boundaries from Web user logs. In *Proceedings of BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research*, pages 57–66, 2000.
[16] He, Q., et al. Web query recommendation via sequential query prediction . *To appear in ICDE'09*, 2009.
[17] Jones, R., et al. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *CIKM'08*, pages 699–708, 2008.
[18] Kang, I., et al. Query type classification for web document retrieval. In *SIGIR'03*, pages 64–71, 2003.
[19] Lafferty, J.,D., et al. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *ICML'01*, pages 282–289, 200.
[20] Lee, U., et al. Automatic identification of user goals in web search. In *WWW'05*, pages 391–400, 2005.
[21] Li, X., et al. Learning query intent from regularized click graphs. In *SIGIR'08*, pages 339–346, 2008.
[22] Mccallum, A. Efficiently inducing features of conditional random fields. In *UAI'03*, pages 403–410, 2003.
[23] Olsson, J.,S., et al. Cross-language text classification. In *SIGIR'05*, pages 645–646, 2005.
[24] Rose, D.E., et al. Understanding user goals in web search. In *WWW'04*, pages 13–19, 2004.
[25] Salton, G., et al. A vector space model for automatic indexing. . In *Communications of the ACM. vol. 18*, pages 613–620, 1975.
[26] Settles, B. Abner: an open source tool for automatically tagging genes, proteins, and other entity names in text. In *Bioinformatics.21(14)*, pages 3191–3192, 2005.
[27] Shen, D., et al. Building bridges for web query classification. In *SIGIR'06*, pages 131–138, 2006.
[28] Sutton, C., et al. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *ICML'04*, 2004.
[29] Wen, J., et al. Clustering user queries of a search engine. In *WWW'01*, pages 162–168, 2001.