Context-Aware Online Commercial Intention Detection

Derek Hao Hu[1,⋆], Dou Shen[2], Jian-Tao Sun[3], Qiang Yang[1], and Zheng Chen[3]

[1] Hong Kong University of Science and Technology
{derekhh,qyang}@cse.ust.hk
[2] Microsoft Research
doushen@microsoft.com
[3] Microsoft Research Asia
{jtsun,zhengc}@microsoft.com

Abstract. With more and more commercial activities moving onto the Internet, people tend to purchase what they need through Internet or conduct some online research before the actual transactions happen. For many Web users, their online commercial activities start from submitting a search query to search engines. Just like the common Web search queries, the queries with commercial intention are usually very short. Recognizing the queries with commercial intention against the common queries will help search engines provide proper search results and advertisements, help Web users obtain the right information they desire and help the advertisers benefit from the potential transactions. However, the intentions behind a query vary a lot for users with different background and interest. The intentions can even be different for the same user, when the query is issued in different contexts. In this paper, we present a new algorithm framework based on skip-chain conditional random field (SCCRF) for automatically classifying Web queries according to *context-based online commercial intention*. We analyze our algorithm performance both theoretically and empirically. Extensive experiments on several real search engine log datasets show that our algorithm can improve more than 10% on F1 score than previous algorithms on commercial intention detection.

1 Introduction

The rapid development of World Wide Web has impacted almost every aspect of our daily life and more and more activities happen on the Internet. Among these activities, one important kind is commercial activities, which form an ecosystem and attract a lot of players.In this ecosystem, the behaviors of Web users play a critical role. The behaviors include shopping online, or conducting online research for actual deals. As we are aware of, most Web users start their online behaviors by submitting a Web query to a search engine. Therefore, accurately understanding the intentions behind the issued queries is of great importance to

[⋆] This work was done when Derek Hao Hu was an intern at Microsoft Research Asia.

Z.-H. Zhou and T. Washio (Eds.): ACML 2009, LNAI 5828, pp. 135–149, 2009.

the mentioned ecosystem. In this paper, we focus on detecting the commercial intentions of Web queries, which is not thoroughly studied yet as the general query intentions studied in [2,11,9,6].

Detecting Online Commercial Intention (OCI) from Web queries is not trivial, considering the following three difficulties. The first difficulty is that many queries are very short. [5] studied an Excite search-service transaction log and showed that approximately 93% of the Web queries contained less than 4 terms. It is extremely hard to derive user intention solely based on the queries. The second difficulty is that a Web query often has multiple meanings and hence is ambiguous. For example, the word "jaguar" has dozens of meanings, which can either mean an animal or a kind of luxury cars, or others[1]. The third difficulty is that the intention of a Web query can vary given different contexts. For example, even if "jaguar" takes the meaning of being a kind of luxury car, it either encodes a commercial intention (when the user wants to buy a car), or non-commercial intention (where the user just wants to find some luxury car pictures).

[4] first defined the notion of OCI and provided a non-context-aware approach to detect OCI in Web queries. The authors formalize the problem as a binary classification problem to decide whether a search query is intended for commercial purposes such as intending to buy a product or finding product information as in the research stage. The proposed solution, based on query enrichment through search engines and traditional text classification techniques [4], solves the first difficulty as we discussed. However, their method cannot tackle the second and third difficulties. In this paper, we propose a new algorithm to analyze the commercial intention of Web queries, taking the contextual information of the submitted queries into consideration. With these information provided, our method can provide more accurate predictions for commercial intention of Web users.

Figure 1 shows the workflow of our algorithm. We consider a newly asked query and then consider two kinds of features, one is the generalized OCI intention degree which extracts features from top result pages when this query is issued to the search engine. The other is the historical similarity feature, which takes past queries into consideration. It first detects all the similar queries in the user's personal query log up to a specific length. And then it computes the semantic similarity kernel function by using query expansion techniques as external information sources. Next these two feature functions are used in a skip-chain conditional random field model (SCCRF). A skip-chain conditional random field model adds possible links between non-adjacent nodes, in contrast to the more commonly used linear chain conditional random field, where only adjacent nodes are connected. The reason for us to use SCCRF is to grab the connection between the query labels (commercial or non-commercial) of non-adjacent nodes since simply connecting edges between adjacent nodes would not accumulate enough information. Finally, the queries are classified as being commercial or non-commercial. Details are presented in Section 3.

[1] http://en.wikipedia.org/wiki/Jaguar_%28disambiguation%29

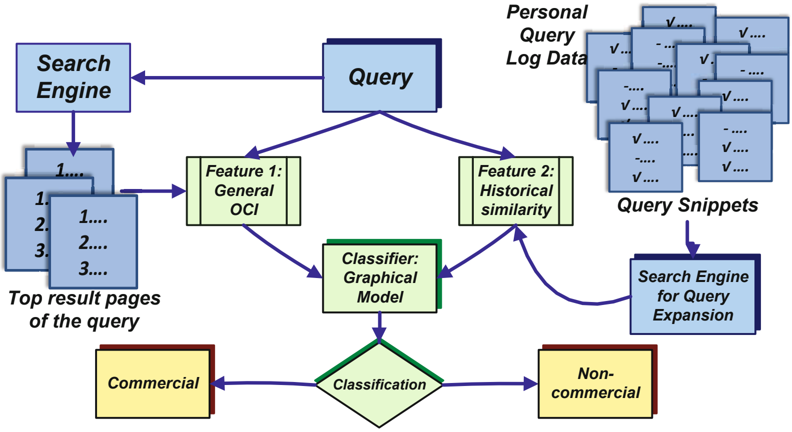
Personal Query Log Data
Search Engine
Query
Top result pages of the query
Feature 1: General OCI
Feature 2: Historical similarity
Query Snippets
Classifier: Graphical Model
Search Engine for Query Expansion
Commercial
Classification
Non-commercial

Fig. 1. Framework of our context-based OCI detection algorithm

Detecting OCI from contextual queries may appear to be similar to the context-aware query classification problem recently proposed in [3]. However, how to integrate contextual information in OCI detection is completely different from context-aware query classification. In context-aware query classification, one can learn much knowledge from direct relationship between adjacent queries as well as the QC taxonomy information. In OCI detection, we can only have limited knowledge from adjacent query relationships since we only have two categories. Furthermore, we do not have a taxonomy from whose structure we can mine useful information. Therefore, we proposed a different approach from [3] to define context-aware features and accumulate useful information from the surrounding queries.

The rest of the paper is organized as follows. We show some related works on general OCI detection and query classification methods. We then present our solution in Section 3 for detecting context-based OCI. In Section 4, we compare our approaches against the baseline method in [4] using real query log data from AOL and Live Search. Finally, in Section 5, we give the conclusion of this paper and describe some possible future research directions.

2 Related Work

A machine learning-based approach for predicting OCI based on Web pages or queries was proposed in [4]. When detecting OCI from Web pages, the traditional approach of document classification is used, where we are given a training set D consisting of training Web pages $\langle d_j, C_i \rangle$, $i = 1$ or 2 and $1 \leq j \leq |D|$, where C_1 means the Web page has commercial intention and C_2 means the Web page has non-commercial intention. The Web pages are represented by the Vector Space Model. The keywords are extracted from both the content texts and tag attributes of all the labeled Web pages in the training data.

The problem of Web query classification is closely related to the OCI detection problem, although current research works on query classification do not tackle this problem. Currently, works on query classification can be split into two groups, one is classifying queries according to query types, such as informational or navigational or transactional[2,7]; and the other is classifying queries according to the query topic, such as "computers/hardware" or "computers/software" [1]. However, as mentioned in [4], OCI follows an independent dimension compared to query topic classification or query type classification. Therefore, the methodologies for these two types of query classification can not be used directly for OCI detection.

One paper that is particularly related to this work is context-aware query classification [3]. In that paper, the authors present a query classification approach also based on conditional random fields and aiming at mining useful information from user sessions. Our work differs from that paper in several aspects.

In [3], the authors defined several kinds of "context-aware" features based on direct association between adjacent labels, and also taxonomy-based association between adjacent labels. However, when we aim to detect online commercial intention, such features cannot be simply borrowed to work in this scenario.

Firstly, in [3], they use a linear-chain conditional random field model and simply calculate the number of occurrences of a pair of adjacent query labels as the direct association between adjacent labels. It's easy to see that in the problem of OCI detection, we can only form four pairs of adjacent query labels and therefore the knowledge we can gain from using such information is very limited. To solve this problem, we propose to apply the skip-chain conditional random field to better capture the information hidden between nonadjacent, but related queries and we will also demonstrate its usefulness in the experimental results section.

Secondly, another part of contextual features is the taxonomy-based association, where the structure of the taxonomy is exploited to mine useful information between "sibling" categories. In particular, some transitions between categories may not occur in the training data and therefore simply using the number of observed transitions between adjacent query labels may not reflect the distribution in real-world applications. Therefore, in [3], the association between two sibling categories are considered to be stronger than non-sibling categories. Nevertheless, in the problem of OCI detection, such a taxonomy does not exist since we only have two categories instead of the multiple categories available in general QC. We also cannot aim to use taxonomy based association to provide contextual features.

Therefore, both the two parts of contextual features proposed in [3] cannot be applied to the OCI detection problem, which makes our problem and approach highly different from [3]. Furthermore, in our experiments section, we do not use any information about the user clicked URLs since that information is missing from our query log data. Such information is available and is defined as a major part of "context" in [3], which further shows the differences between our work and [3].

3 Context-Based OCI Detection

3.1 Overview

In this section, we describe our algorithm for context-based online commercial intention detection from personal query logs.

We first make the basic assumption on context-based OCI detection: a search engine has access to the query log of a specific user, or at least the query log from this same IP address. Here we assume that a query log (or clickthrough log in some literature) consists of a set of queries associated with the user-clicked search result pages or snippets. Stated formally, the query log is a set Q where each element is at least a triple $\langle U, T, Q, [C]\rangle$, where U indicates the user ID, or any other information (e.g. IP address) that can differentiate one user from another, T indicates the time where this query is issued and Q indicates the string of the Web query. Other elements may also be added to this query log so that more information will be encoded, such as in the case of a clickthrough log where we have C, which indicates what pages or URLs the Web user actually clicked on or the snippets of the clicked pages. In the following, we refer to the log data consistently as query logs, and do not consider the existence of C since including these contents would be straightforward.

The assumption on user identification can often be satisfied in real world, where search engine companies record the query logs of different registered users or their IP addresses for differentiating them from each other. When the personal query log data are available, for each incoming query, we can use the information from the personal query log to find similar queries that has strong correlation with the new query by the same user or user group.

3.2 Modeling Query Logs via CRF

Since we want to take a sequence of queries instead of a single query as our input when we train the classifier, we find that the problem fits well with conditional random field as our graphical model for context-based OCI detection. Conditional Random Field, which was first proposed by Lafferty et al.[8], is widely used in relational learning which directly models the conditional distribution $p(\mathbf{y}|\mathbf{x})$. In this paper, we use a variant of the widely used linear-chain CRF model, the skip-chain CRF proposed in [12], to model the context-based OCI issue for the following reasons. Firstly, skip-chain CRF has deep roots in natural language processing area (NLP). In NLP, the problem of Named Entity Recognition (NER) has similarities with the context-based OCI detection problem, which needs to model the correlation between non-consecutive identical words in the text. Secondly, being a probabilistic graphical model, skip-chain CRF has its advantage in modeling uncertainty in a natural and convenient way. Thirdly, the key issue in skip-chain CRF is how to add skip edges. Semantic similarities, which represents how similar two queries are in terms of their intended meaning such as category of targeting pages or products, are the major issues we consider when creating skip edges in the CRF model. Based on the above reasons, we believe that skip-chain CRF would be a model appropriate for handling context-based OCI detection.

The main advantage of a skip-chain CRF (SCCRF) model over the commonly used linear-chain CRF models is that the skip-chain CRF model has an additional type of potential, which is represented using long-distance edges. Formally, we build a SCCRF model as follows. Assume that the original personal query log length is N. In order to acquire training data consisting of personal query logs with each length as L, we can build a training set with cardinality $(N - L + 1)$, where each data instance consists of L consecutive queries in the original personal query log, i.e. each personal query log starts with the query item $1, 2, \ldots, N - L + 1$ and has a length of L. In our experiments of Section 4, we will empirically evaluate the classifier performance with different values of L.

Graphical models like CRF or skip-chain CRF directly represents the conditional distribution $p(\mathbf{y}|\mathbf{x})$, where in our context-based OCI detection problem, $\mathbf{y}$ indicates the target label of each query as being commercial or non-commercial and $\mathbf{x}$ states an observed personal query log of length L. We let x_t be the observed t^{th} query in the personal query log, and let y_t be a random variable to indicate the OCI value inferred from the t^{th} query, in the final setting if the y_t inferred is larger than 0.5, we assume that the given query has commercial intention. This parameter setting of threshold follows that of [4].

The conditional random field model is represented by a factor graph, which is a bipartite graph $G = (V, F, E)$ in which a variable $v_s \in V$ is connected to a factor node $\Psi_A \in F$ if v_s is an argument to Ψ_A. For skip-chain CRF, it is essentially a linear-chain CRF with additional long-distance edges between queries x_i and x_j such that $f(x_i, x_j) > \theta$ (Refer to Figure 2 for an illustration). θ is a parameter that can be tuned to adjust the confidence of such correlations between different queries. In our experiments, we will evaluate the effect of changing the parameter θ on the classifier accuracy.

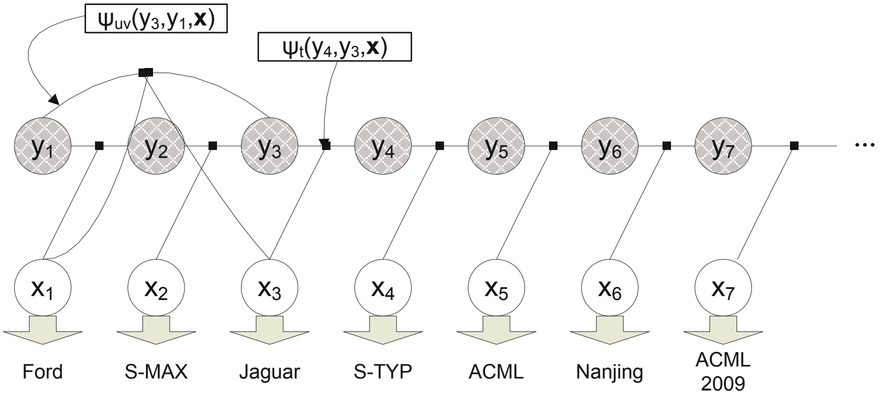

Fig. 2. Illustration of the SCCRF model

For an observation sequence $\mathbf{x}$, let $\mathcal{I} = \{u, v\}$ be the set of all pairs of queries for which there are skip edges (all edges except edges connecting adjacent queries) connected with each other. The probability of a label sequence $\mathbf{y}$ given an observation activity sequence $\mathbf{x}$ is:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{t=1}^{n} \Psi_t(y_t, y_{t-1}, \mathbf{x}) \prod_{(u,v)\in I} \Psi_{uv}(y_u, y_v, \mathbf{x}). \tag{1}$$

In the above Equation 1, Ψ_t are the potential functions for linear-chain edges and Ψ_{uv} are the factors over the skip edges (Also refer to Figure 2 for illustration). $Z(x)$ is the normalization factor. We define the potential functions Ψ_t and Ψ_{uv} in Equation 2 and Equation 3 as:

$$\Psi_t\left(y_t, y_{t-1}, \mathbf{x}\right) = \exp\left(\sum_k \lambda_{1k} f_{1k}\left(y_t, y_{t-1}, \mathbf{x}, t\right)\right) \tag{2}$$

$$\Psi_{uv}\left(y_u, y_v, \mathbf{x}\right) = \exp\left(\sum_k \lambda_{2k} f_{2k}\left(y_u, y_v, \mathbf{x}, u, v\right)\right) \tag{3}$$

λ_{1k} are the parameters of the linear-chain template and λ_{2k} are the parameters of the skip-chain template. Each of the potential functions factorize according to a set of features f_{1k} or f_{2k}. We will describe our choice of feature functions later.

Learning the weights λ_{1k} and λ_{2k} for the skip-chain CRF model can be achieved by maximizing the log-likelihood of the training data, which requires the computation of calculating partial derivative and optimization techniques. We omit the detailed algorithm of inference and parameter estimation of the skip-chain CRF model. Interested readers can consult [8,12] for technical details.

3.3 Modeling Semantic Similarities between Queries

We now specify how to calculate the feature functions in our CRF settings. Two kinds of feature functions must be computed. One is the “query-intention” pair, the other is the “query-query” pair. For the first function, it is easy to follow the traditional IR techniques, where we can first choose some keywords from the query snippets or the top landing pages, count the occurrence of these words and then learn the corresponding weight. In this paper, we used the OCI value of top 10 result pages calculated from the baseline method[2] [4] as the features representing the “query-intention” pair.

The remaining issue is how to define a good “query-query” similarity function that can measure the semantic similarity of different search queries. An accurate function that can reflect the inherent semantic similarity or correlation between

[2] http://adlab.msn.com/OCI/OCI.aspx

different queries will substantially improve the overall accuracy. Due to the inherent feature of the length of queries, the traditional method of word-based similarity metric cannot be used. Therefore, we use an approach, based on the idea of query expansion, to measure the similarity of short text snippets the considered query pairs.

Our first expansion, called the "first-order" query expansion, is to compute the cosine similarity of query snippets. Given two queries Q_1 and Q_2, we first issue these two queries into the search engine and then retrieve the result pages of the corresponding two queries and the m query snippets. Then we combine these m query snippets as one document and consider the cosine similarity between them. In other words, we first compute the TFIDF vector of the two documents, denote them as A and B, and then compute: $\theta = \arccos \frac{A \cdot B}{\|A\| \|B\|}$.

However, although cosine similarity is a traditional similarity metric, it may not satisfy our need to measure the semantic similarity between different queries since the query snippets are rather short and the document pairs will not contain common terms. Also even if we consider the cases of the two snippets containing the same terms, it may not mean that these same terms mean the same thing in different contexts of different query snippet documents. So measuring similarity based on word terms is not a good choice for our problem. In our experiment section 4, we will show that our "first-order" query expansion does not perform very well, compared to the "second-order" query expansion we used.

Our second expansion goes a step further, compared to the "first-order" query expansion, and we call the idea of "second-order" query expansion based on the pages we get when we issue the query snippets again into the search engine. In this way, the information we get from this particular query snippet is increased. This step is similar to the solution given in [10]. We consider the maximum similarity between these query snippets and take the value as the value of feature function between "query-query" pairs in the SCCRF model.

Stated formally, we have $f(y_u, y_v, x) = \max_{1 \le i \le m, 1 \le j \le m} g(S_{ui}, S_{vj}))$, $g(S_{ui}, S_{vj})$ is defined as the value of similarities between query snippet S_{ui} and S_{vj}, where S_{ui} is the i^{th} query snippet when we issue the u^{th} query into the search engine, S_{vj} is defined similarly.

We use a kernel function to compute the semantic similarities of given query pairs based on the query expansion framework. Let S_x represent a query snippet. First, we get more expanded information of S_x by using the idea of query expansion. We input this query snippet S_x into the search engine and the top n returned Web pages are retrieved, say, $p_1, p_2, \ldots, p_n$. Then we compute the TFIDF term vector v_i for each Web page p_i. For each v_i, it is truncated to only include its m highest weighted terms, where $m = 50$, is used as a balance between evaluation efficiency and expressive power.

Then we let $C(x)$ be the centroid of L_2 normalized vectors: $C(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{v_i}{\|v_i\|_2}$. Finally, we compute $QE(x)$, $QE(x)$ is the L_2 normalization of $C(x)$: $QE(x) = \frac{C(x)}{\|C(x)\|_2}$. The kernel function of query snippets $K(x, y) = QE(x) \cdot QE(y)$. It can be observed that $K(x, y)$ is a valid kernel function.

Therefore, after defining the corresponding feature functions between "query-intention" pairs and "query-query" pairs, we have enough information to build the SCCRF model from the training data. When testing, a new query arrives, and we take the past $L-1$ query into consideration, which forms together a personal query log with L subsequent queries, and then label the query sequence $\mathbf{y}$ and take the last element from the vector y_L as the label: commercial / non-commercial of this query. We further describe our algorithm workflow in Algorithm 1.

Algorithm 1. Algorithm description for context-based online commercial intention

Input: N is the length of a query log, where each query item is represented by $\{x_i, y_i\}$, where x_i is the i^{th} query and y_i is the corresponding i^{th} label for x_i. Q, which is a newly asked query.

Output: P, which is the probability for Q as being commercial intended.

Assumption: Assume all the queries in the personal query log we considered here are issued by the same user or user group.

Parameters: θ and L, which suggests the confidence parameter for us to add the skip edges and the length of the personal query log training data, correspondingly.

```
for $i = 1$ to $N - L + 1$ do
    Initialize the $i^{th}$ training data as empty.
    for $j = 0$ to $L - 1$ do
        Add the $(i + j)^{th}$ query $x_{i+j}$ to the $i^{th}$ training data.
    end for
end for
for $i = 1$ to $N$ do
    Issue the query $x_i$ to the search engine to get the top $P$ landing pages. $P$ can
    be tuned to reflect more information from landing pages. To simplify, we set
    $P = 10$ in our experiments.
    Compute the corresponding OCI value of these landing pages from the baseline
    method [4].
    Use these values as features for $f_1$.
end for
Train the corresponding SCCRF model from the training set created.
for $i = N - L + 2$ to $N$ do
    Add the query $x_i$ to the test personal query log.
end for
Add the query $Q$ to the test personal query log. Now it contains $L$ terms.
for $i = 1$ to $L$ do
    for $j = 1$ to $i - 1$ do
        Compute the semantic similarity of $T_i$ and $T_j$, i.e. $K(T_i, T_j) = QE(T_i) \cdot$
        $QE(T_j)$ as defined.
        if $K(T_i, T_j) > \theta$ then
            Add a skip edge between $y_i$ and $y_j$, corresponding to the feature
            function $f_2(y_i, y_j, \mathbf{x})$.
        end if
    end for
end for
```

4 Empirical Evaluation

In order to validate the effectiveness of our algorithm, we compare our algorithm to the baseline method proposed in [4]. Several parameters occur in our algorithm, θ, which appears in the skip-chain Conditional Random Field model to mark the confidence of the nonadjacent edges we created. The larger the θ is, we are more confident of the semantic similarity of the edges. Furthermore, another parameter is the parameter L, which is the length of each personal query log in the training data set, also it means when two queries have a distance more than L, we will not consider their semantic similarities because otherwise if the two queries have distances larger than L, they may not be in the same searching session and have no temporal correlation between each other, even though they may have high semantic similarities. In our experiment, we will empirically evaluate how our classifier performance will be affected when we tune these two parameters.

4.1 Description of Datasets

We use two datasets in our experiment, the first is a publicly released AOL query log dataset[3]. The AOL query log data consists of around 20M Web queries collected from around 650,000 Web users, where the data is sorted by anonymous User ID and sequentially arranged. Another query log data we acquired is from Live Search and the original query log data is collected in March 2008. The Live Search query log data consists of around 450M distinct web queries from around 2.5M different query search sessions.

The dataset format of the two querylog datasets are rather similar. Each item of the query log datasets includes {AnonID, Query, QueryTime, ItemRank, ClickURL}. AnonID is an anonymous user ID number. Query is the query issued by the user, case shifted with most punctuation removed. QueryTime is the time at which the query was submitted for search. ItemRank was the rank of the item on which they clicked, if the user clicked on a search result. ClickURL is the domain portion of the URL in the clicked result, if the user had clicked on a search result.

In this paper we do not use the clicked URL information (ClickURL), since this information is often relatively sparse in the query log. Another reason for us not to consider clickthrough information in computing the semantic similarities of the SCCRF model is the infeasibility of getting the clicked URL in most cases. For example, in the AOL query log dataset we only have access to the clicked domains, not the clicked webpages. Therefore we settled on expanding the query information by a “second-order” expansion instead.

Because both the AOL dataset and the Live Search dataset are rather huge and do not contain any label of commercial intention, we had randomly chosen 100 users who had submitted at least 100 queries from both AOL and Live Search datasets and had manually labeled 100 consecutive queries for each of the 100

[3] http://www.gregsadetsky.com/aol-data/

users we had selected. Three labelers labeled the 20K queries we had chosen. Each labeler is told to take the surrounding queries as well as clicked URLs into account to determine the commercial intention degree of the labeled query. Each query is labeled as being "commercial", "non-commercial" or "unable to determine". A query is labeled as "commercial" if and only if at least two out of three labelers mark it as "commercial" and it's similar for "non-commercial" queries. We also delete some invalid queries in the query log. The distribution of commercial intention in the queries of the Web users we've chosen above is shown is the following Table 1. In all we had acquired 9,553 queries in AOL dataset, where 1,247 queries are labeled as commercial and the rest labeled as non-commercial; also we labeled 9674 queries in Live Search dataset, where 936 queries are labeled as commercial and 8,738 queries as non-commercial.

Table 1. OCI distribution of the selected datasets

Labeler	AOL Commercial	AOL Non-commercial	Live Commercial	Live Non-Commercial
1	1238	8627	919	8819
2	1430	8435	1025	8713
3	1117	8748	973	8765
Sum	1247	8306	936	8738

In the query log data, we first performed preprocessing to remove all invalid queries. We then divide each user's query into ten pairs of training and test data, by first choosing a random number between a certain size interval as the size of the training data while keeping the rest of the data for the user as the test data. We repeat this process ten times so as to obtain average results. For all the experiments in this section, we use precision, recall and F1-measure as the evaluation metric.

4.2 Performance of Baseline Classifier

For the baseline method, we use the classifier which is now currently available on the Web[4], following the work in [4], The parameter chosen in the Website is the best-tuned so we just compare the performance of our algorithm with this result. The classification result is in the following Table 2.

Table 2. Baseline Classifier Performance

Dataset	Precision	Recall	F1-Measure
AOL	0.817	0.796	0.806
Live Search	0.802	0.836	0.809

[4] http://adlab.msn.com/OCI/OCI.aspx

4.3 Varying the Confidence Parameter θ

We then analyze how different parameters of θ and l will affect our algorithm performance. When we vary the confidence paramter θ, another important objective is to verify the usefulness of skip edges created between non-adjacent queries and that skip-chain CRF beats the linear chain CRF in the OCI detection scenario.

We first set $l = 1000$ and tune different parameters on θ, we get the following result in Table 3. We run the experiments 10 times with different values of p, accuracy and variance are recorded in the following table. Column with header "AOL (Variance)" is the accuracy and variance of our algorithm performance on our selected AOL query log dataset and "Live Search (Variance)" is the accuracy and variance of our algorithm performance on Live Search dataset. Here we first set L, which is the length of the CRF model, as 50.

Table 3. Algorithm performance with varying parameter θ

θ	AOL (Variance)	Live Search (Variance)
$\theta = 0.01$	0.863 (0.002)	0.872 (0.003)
$\theta = 0.02$	0.887 (0.005)	0.878 (0.003)
$\theta = 0.04$	0.892 (0.003)	0.881 (0.004)
$\theta = 0.08$	0.901 (0.005)	0.893 (0.002)
$\theta = 0.1$	**0.913 (0.002)**	0.901 (0.004)
$\theta = 0.2$	0.912 (0.005)	**0.908 (0.003)**
$\theta = 0.4$	0.902 (0.004)	0.883 (0.006)
$\theta = 0.8$	0.871 (0.003)	0.852 (0.008)
Baseline	0.806	0.809

From Table 3, it is noteworthy to see that our proposed algorithm for context-based online commercial intention *always* performs better than the baseline approach, suggesting that taking contextual information, especially surrounding queries in a query session would possibly help. We can see that for different user queries, different values of θ may lead to best generalization ability. Also it's reasonable that the generalization ability will be best when θ is neither too big nor too small.

When θ is too small, different queries that may not be so relevant will be linked towards each other and hence noise is added to the edges between "query-query" pairs. When θ is large enough, classification accuracy will drop rapidly. This is due to the fact that large values of θ will be a too strict criteria between different queries, in that merely no skip link will be created. Therefore, large values of θ would create too few skip edges and then we cannot model the interleaving processes of user search behaviors through a CRF model which is very similar to a linear chain. This verifies our claim in Section 2 that using a linear-chain conditional random field and simply model the relationship between adjacent queries cannot effectively model the commercial intention of queries as a skip-chain CRF will do.

4.4 Varying Training Data Length L

In the next experiment, we test whether different parameters of L will lead to large variance in classification accuracy. From the result in the earlier experiment, here we empirically set θ as 0.1. We get the following result in Table 4.

Table 4. Algorithm performance with varying parameter L

L	AOL (Variance)	Live Search (Variance)
$L = 5$	0.872 (0.010)	0.871 (0.013)
$L = 10$	0.893 (0.011)	0.878 (0.010)
$L = 15$	0.882 (0.009)	0.891 (0.005)
$L = 20$	0.901 (0.005)	0.891 (0.003)
$L = 25$	0.910 (0.004)	0.897 (0.007)
$L = 30$	**0.913 (0.002)**	0.901 (0.004)
$L = 40$	0.909 (0.003)	**0.903 (0.005)**
$L = 50$	0.905 (0.003)	0.902 (0.003)
Baseline	0.806	0.809

Again, our experimental results show that the algorithm performance is rather steady although it typically suggests that it's better to set the history length L at around 30, which would achieve the best performance so far. And even if we only set L as 5, i.e. consider the history submitted queries up to 5, it would still perform better than the baseline approach.

4.5 Analysis of Training Time

In this subsection we will show the training time of our proposed approach with varying lengths of L. Since we have calculated the accuracy as well as variance from a "ten-fold wise" appoach, we would show the total time for us to train the model. Thus, the average time to train a CRF model of length L would be the time shown divided by 10. The result is shown in the following Table 5. Time is measured in seconds.

The result is rather promising. Even when we consider history length up to 50, the computational time is rather quick, only around 15 seconds to train a CRF model for ten times. Such a model is undoubtedly a good fit for real-world usage.

4.6 Comparison between Query Expansion Methods

Finally, we verify our claim that our way of "second-order" query expansion will perform better than "first-order" query expansion. We choose the settings $\theta = 0.1$ and $L = 30$ in "second-order" query expansion and test them against the "first-order" query expansion. The result is shown in Table 6.

Table 5. Training time with varying lengths of L on a Pentium Core 2 Dual 2.13GHz CPU

L	AOL Time	Live Search Time
$L = 5$	1.7s	1.7s
$L = 10$	3.0s	4.1s
$L = 15$	4.9s	5.2s
$L = 20$	6.2s	6.8s
$L = 25$	9.0s	10.2s
$L = 30$	11.1s	11.7s
$L = 40$	14.0s	14.1s
$L = 50$	15.3s	16.3s

Table 6. Comparison of first-order query expansion vs. second-order query expansion

Dataset	Baseline	First-Order	Second-Order
AOL	0.806	0.825 (0.007)	0.913(0.002)
Live Search	0.809	0.826 (0.006)	0.901(0.004)

The above table shows our use of "second-order" query expansion can substantially improve the quality over "first-order" query expansion alone. However, due to the fact that computing the semantic similarities through the "second-order" query expansion approach might be slow. It would be a tradeoff to use first-order query expansion instead, i.e. decrease the computational time while sacrifice some prediction accuracies.

5 Conclusion

In this paper, we have presented a new algorithm based on contextual information to solve the problem of context-based online commercial intention detection. Our work follows from the intuitive motivation that in the same session of a personal query log, semantic similarities and the correlation between surrounding queries can help improve the overall classification accuracy. Similar assumptions are also made in [3]. However, the context-aware features defined in that paper cannot be simply borrowed to tackle the OCI detection problem. Therefore, we exploited a skip chain CRF model to model the problem as a collective classification problem. We use an algorithm based on query expansion to consider the semantic similarity between queries, using query snippets as a major source of information. Our experiment on the AOL query log dataset as well as the Live Search Clickthrough Log Dataset shows that our algorithm can effectively improve the accuracy of context-based OCI detection.

Acknowledgement

Qiang Yang and Derek Hao Hu thank the support of Microsoft Research Project MRA07/08.EG01.

References

1. Beitzel, S.M., Jensen, E.C., Frieder, O., Grossman, D.A., Lewis, D.D., Chowdhury, A., Kolcz, A.: Automatic web query classification using labeled and unlabeled training data. In: SIGIR 2005, pp. 581–582 (2005)
2. Broder, A.Z.: A taxonomy of web search. SIGIR Forum 36(2), 3–10 (2002)
3. Cao, H., Hu, D.H., Shen, D., Jiang, D., Sun, J.-T., Chen, E., Yang, Q.: Contextaware query classification. In: SIGIR 2009 (2009)
4. Dai, H.K., Zhao, L., Nie, Z., Wen, J.-R., Wang, L., Li, Y.: Detecting online commercial intention (oci). In: WWW 2006, pp. 829–837 (2006)
5. Jansen, B.J.: The effect of query complexity on web searching results. Information Research 6(1) (2000)
6. Jansen, B.J., Booth, D.L., Spink, A.: Determining the user intent of web search engine queries. In: WWW 2007, pp. 1149–1150 (2007)
7. Kang, I.-H., Kim, G.-C.: Query type classification for web document retrieval. In: SIGIR 2003, pp. 64–71 (2003)
8. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML 2001, pp. 282–289 (2001)
9. Li, X., Wang, Y.-Y., Acero, A.: Learning query intent from regularized click graphs. In: SIGIR 2008, pp. 339–346 (2008)
10. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: WWW 2006, pp. 377–386 (2006)
11. Shen, D., Sun, J.-T., Yang, Q., Chen, Z.: Building bridges for web query classification. In: SIGIR 2006, pp. 131–138 (2006)
12. Sutton, C.A., Rohanimanesh, K., McCallum, A.: Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In: ICML 2004 (2004)