

An algorithm for analyzing personalized online commercial intention

Derek Hao Hu[†], Qiang Yang[†], Ying Li[‡]

[†]Department of Computer Science and Engineering,
Hong Kong University of Science and Technology
{derekhh, qyang}@cse.ust.hk

[‡]Microsoft Corporation
One Microsoft Way, Redmond, WA, USA
yingli@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 deals 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. The only existing research work, as far as we know, has been done to automatically detect online commercial intention purely based on the issued queries, without considering the Web user's information. 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 novel algorithm, which we name as **POINT**, for the Personalized Online-commercial INTention detection based on a skip-chain conditional random field model. To accurately detect the commercial intentions of a query, our method comprehensively considers the evidences from the target query, the profile of the user issuing the query, which is inferred from his search history, as well as the similarity of different queries in a personal query log. Our proposed method is validated through extensive experiments on a real search engine query log data set. The experimental results show that our algorithm can clearly improve the performance by more than 10% of personalized online-commercial intention detection.

Categories and Subject Descriptors

H.3.m [Information Storage and Retrieval]: Miscel-

laneous; I.5.2 [Pattern Recognition]: Design Methodology—*classifier design and evaluation*

General Terms

Algorithms, Experimentation

Keywords

Online-commercial Intention, Conditional Random Field, semantic similarity

1. INTRODUCTION

The rapid development of World Wide Web has impacted almost every aspects of our daily life. As a result, 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. For example, Web search engines provide the right information to Web users; advertisers invest in online advertising for potential transactions; publishers provide decent content to attract Web users and advertisers. In this ecosystem, the behaviors of Web users play a critical role. The behaviors include shopping online, or conduct 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 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 [3, 15].

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. [8] 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 others¹. 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

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

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).

As far as we know, [5] is the only work focusing on commercial intention detection 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 is actually based on query enrichment through search engines and traditional text classification techniques [5], which solves the first difficulty as we discussed. However, their method cannot tackle the second and third difficulties. In this paper, we put forward a novel solution, called **POINT**, to handle these difficulties by considering more information of Web users, as well as their search histories. With these information, our method can potentially provide more personalized and focused predictions for user intention.

Our proposed method is inspired by the observations that the available query logs contain much Web users' search behavior and have valuable hidden knowledge that can be used to infer Web users' intention. A query log typically consists of User ID, the time at which the query was issued and the query string. If we extract the query strings issued by the same user, then we call the log we extracted as "personal query log". If the query log also contains clickthrough URLs or pages, then we call the augmented query log "clickthrough log".

From the personal query logs, we build a probabilistic graphical model based on a user's historical queries, which can act as the user's personal profile. Then, given a new query, we use a kernel function to compute the semantic similarity between different queries, thereby drawing collective inferences on the new query's intention based on the users' previous similar queries. We evaluate our proposed algorithm **POINT**, for personalized OCI using a real Web search query log. The experiments demonstrate that our method, using personalized information and historical queries, can significantly improve the OCI accuracy.

In our algorithm, we consider a newly asked query and then consider two kinds of features, one is the generalized OCI intention 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 firsts detects all the similar queries in the user's personal query log. And then it computes the semantic similarity kernel function by using a "second-level" of query expansion. Next these two feature functions are used in a conditional random field model. Finally, the query data are classified as being commercial or non-commercial. Details are presented in Section 3.

The rest of the paper is organized as follows. We show some related works on general OCI detection, query classification and personalized search in Section 2. We then present our solution in Section 3 for detecting personalized OCI. In Section 4, we compare our approaches against the baseline method in [5] using a real query log data publicly released by a search engine company. Finally, in Section

5, we give the conclusion of this paper and describe some possible future research directions.

2. RELATED WORK

One direct related work to this paper is the Online-commercial intention detection in [5], in which the authors detect the commercial intention of a query in a general sense, without considering the contextual information of the query. Another group of related work is query topic or type classification, which faces the same difficulties as we mentioned. Actually, our proposed solution can be used to solve these query classification problems. We also discuss the relatedness of our work with the rapidly progressing field of personalized search, which also tends to detect user interest to provide personalized relevant pages.

2.1 General OCI Detection

A machine learning-based approach for predicting OCI based on Web pages or queries was proposed in [5]. 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 (d_j, C_i) , $i = 1$ 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.

For the feature selection step, terms are selected such that keywords are not only "significant" in that the features can distinguish different class labels, they should also be frequent enough to be reliable and representative. After the keywords are selected, the number of appearances of such keywords in the inner texts and in the tag attributes of Web pages are counted. Content texts and tag attributes are distinguished because the latter tracks the texts on special elements, such as buttons, images and forms, which has a very different role from the general text, especially for commercial detection, e.g. when the word "order" appears on a button it usually implies a very high confidence of commercial intention. Such values are used as final feature representations. After the features are extracted, a traditional SVM algorithm is used for learning a classifier.

When detecting OCI from queries, contents of top landing pages recommended by the search engine are used. This approach is reasonable since top-ranked result pages give a deeper exploration of the user's intention. The content of the first search result page, or the top query snippets, which usually contain title, short descriptions and URL links to the recommended landing pages are used as well. The corresponding OCI confidence values of each of these pages are used as features for training the given query data. Thus, as we can see, [5] follows a classical supervised learning framework to solve the OCI detection problem.

Although the above described method can solve the first difficulty (Sparse information in Web queries) as mentioned in Section 1 by enriching queries through search engines, it cannot solve the second and third difficulties. For example, given the query "jaguar", by query enrichment, we may obtain a list of Web pages talking about either a special

animal, or a luxury car such that we can know this query can potentially reflect a commercial intention. However, we cannot detect whether the user is intended to buy a car or not when he issues this query. This can only be achieved by considering the contextual information of this query such as the history search information from this user. The method proposed in this paper is to exploit such information for this goal.

2.2 Query Classification

The problem of Web query classification is closely related to the OCI detection problem, although the current trend of query classification does 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 [3, 9]; and the other is classifying queries according to the query topic, such as “computers/hardware” or “computers/software”. [2, 15]. However, as mentioned in [5], 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.

For query type classification, Broder classified Web queries according to their intents into three types: informational, navigational and transactional in [3]. The intent for informational queries is to acquire some information assumed to be present on one or more web pages; for navigational queries, it is to reach a particular site; for transactional queries, it is to perform some web-mediated activity. Although Broder did not provide a way to distinguish different types of queries, he makes an informative survey which shows that the navigational queries take up 24.5% of all the queries while the informational queries and transactional queries take up 39% and 36% approximately. Identifying the types of Web queries and providing appropriate search results accordingly is critical for the new generation of search engines. Following Broder’s work, Lee et al. further study whether the types of a query is predictable and how to predict it [11].

For query topic classification, queries are classified into some predefined categories according to query topics or subjects. Earlier work of query topic classification was done by manually classifying Web queries for query analysis, especially on the query topic distribution [1].

Recently, automatic topic classification techniques have been exploited and they have been used for building query filters, study of user interests and enriching Web taxonomies. Typical query classification methods expand a query through search engines which results in a list of related Web pages together with their categories from an intermediate taxonomy. A straightforward approach is to leverage the categories by exact matching. This approach tends to produce classification results with high precision but low recall. Therefore statistical learning methods can be used for query topic classification problems. Queries are enriched by searching related pages which can specify the meanings of queries more accurately. Then the queries are classified based on the trained SVM classifier.

In [16], a bridging classifier on an intermediate taxonomy was created in an offline mode. Then it was used in an online mode to map user queries to the target categories via the intermediate taxonomy. The bridging classifier needs to be trained only once. By leveraging the similarity distribution over the intermediate taxonomy, a new classifier does not need to be retrained for each new set of target categories.

2.3 Personalized Search

Personalized search has been proposed for many years and there have been many current personalization strategies. It is also obviously related to our work, which also attempts to incorporate personalized information into commercial intention detection. Some earlier approaches [4] try to tackle this program by defining different similarity measures between the web pages and user interests, which are required to be specified, either in the form of user profiles or encoded user interests. However, due to the fact that users often do not wish to provide feedback, either in an implicit or explicit way, which can reflect their potential interest or user profiles. Thus, later works tend to focus on automatically learn the user profiles without the effort from the user’s side, e.g. to learn user interest from clickthrough data [13].

Besides these, many algorithms were tuned based on the original PageRank algorithm, for personalized web search based on the hyperlink structures of the Web. In [7], Topic Sensitive PageRank algorithm was proposed, one score assigned to each main topic of ODP. In [11], user goals, or user interests are estimated from the user’s clickthrough histories, based on the Topic-Sensitive PageRank scores of these pages.

Other researchers tend to solve this problem by incorporating the information from a group of users, believing that it would be reasonable to “borrow” the interests from other users that have similar actions or interests with the user we are interested in. Such an idea, commonly known as collaborative filtering, is a typical group-based personalization method and has also been used in personalized search. For example, [17] proposes CubeSVD, which applies personalized search by incorporating the relationships between users, queries, and web pages in the click-through data.

There exist many different personalized search strategies and thus an extensive overview will be beyond the length limit and scope of this paper. Interested readers can refer to [6] for a more extensive overview and comparison of the effectiveness and efficiency of different personalized search strategies.

3. PERSONALIZED OCI DETECTION

In this section, we describe our algorithm for personalized online commercial intention detection from personal query logs. We exploit a graphical model for labeling the personal query log of a given user. Semantic similarities between different queries will be calculated using a kernel function and correlations between nonadjacent queries are taken into consideration.

We first make the basic assumption on personalized 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.

For example, when we submit a query “apple”, the search engine does not know whether the user is intended to search for the fruit or for the computer company. By considering his queries issued in the past of the same session, we know that the user has also searched some computer-related terms in this session. Thus, the probability of this user searching “apple” with his target as the computer company is therefore increased greatly. Another example is the query “jaguar”. Assume that at first a search engine has no idea about the intention of this query, the user may be targeting the animal or the car company. Again, it will be helpful if the queries in the same session contain some other queries related to these two areas, thereby improving the possibility of a specific intention.

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 personalized OCI detection. Conditional Random Field, which was first proposed by Lafferty et al.[10], 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 [18], to model the personalized 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 personalized 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 refers to 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 personalized 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 has 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 $1, 2, \dots, N - L + 1$ and has a length of L . In our experiment 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 personalized 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 [5].

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 1 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.

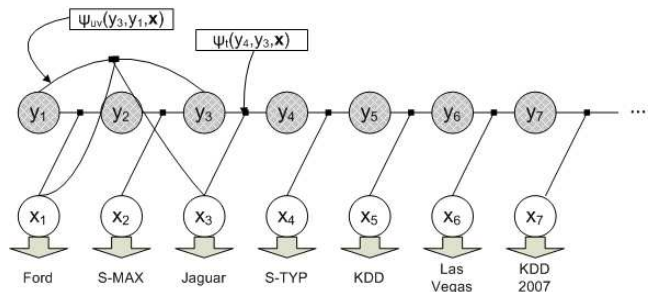


Figure 1: Illustration of the SCCRf model

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 sequence \mathbf{x} is:

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

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 1 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:

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

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

λ_{1k} are the parameters of the linear-chain template and λ_{2k} are the parameters of the skip-chain template. Each of them 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 calculation of calculating partial derivative and optimization techniques. We omit the detailed algorithm of inference and parameter estimation of the skip-chain CRF model.

Exact inference in skip-chain CRFs may be intractable as the time complexity is exponential in the size of the largest clique in the junction tree of the graph, and that there may be long and overlapping loops in the model. Loopy belief propagation (LBP) is used widely for performing approximate inference in CRFs. In a skip-chain CRF, LBP cannot guarantee convergence. However, empirical studies show that LBP has been effective [19, 20]. Due to space constraint, we omit the rather standard steps of learning and inferring in CRF models here. Interested readers can consult [10, 19] for technical details.

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 method² [5] 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 different queries will substantially improve the overall accuracy. Due to the fact that queries are often short in nature, the traditional method of word-based similarity metric cannot be used. An important idea is to expand the queries, or expand the knowledge by the “query expansion” step. Another possible method is to use the clickthrough data of queries, where similar queries will tend to click on similar Web pages or pages that have the same or similar topical categories.

The problem of measuring the semantic similarity between Web search queries by mining the clickthrough data has

²<http://adlab.msn.com/OCI/OCI.aspx>

been considered before. Zhao et al [21] proposed a time-dependent semantic similarity measure of queries, using a marginalized kernel technique. However, since clickthrough data in our experiment is sparse and the Web domain, instead of the actually clicked exact URL is given in the query log, it is impossible for us to exploit the use of clickthrough data and evaluate them in our experiment. Therefore, we use a different approach, based on the idea of query expansion, to measure the similarity of short text snippets the considered query pairs. This idea is also related to the query enrichment framework, which we can map the short query to the intermediate query snippets for future use [15].

Our first expansion, called the “first-level” 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 of them. In other words, we first compute the TFIDF vector of the two documents, and then compute:

$$\theta = \arccos \frac{A \cdot B}{\|A\| \|B\|}. \quad (4)$$

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-level” query expansion does not perform very well, compared to the “second-level” query expansion we used.

Our second expansion goes a step further, compared to the “first-level” query expansion, and we call the idea of “second-level” 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 [14]. 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 SCCR model.

Stated formally, we have:

$$f(y_u, y_v, x) = \max_{1 \leq i \leq m, 1 \leq j \leq m} g(S_{ui}, S_{vj}) \quad (5)$$

$g(S_{ui}, S_{vj})$ 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, \dots, 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.

4. EMPIRICAL EVALUATION

In order to test the effectiveness of our algorithm, we compare our algorithm **POINT**, to that of a general OCI classifier, i.e. the baseline method proposed in [5]. Several parameters occur in our algorithm, θ , which is occurred 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 sure 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 Dataset

Our experiment uses a publicly released query log released by a real Web search engine. The 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.

Each data 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 this dataset we only have access to the clicked domains, not the clicked webpages. Therefore we settled on expanding the query information by a “second-level” expansion instead.

Because the original dataset is rather huge and does not contain any label of commercial intention. We chose four users out of the whole dataset which has substantially more queries than other users. Their user IDs are: 42075, with 5,165 items in the query log; 117514, with 7,545 items in the query log; 2263543, with 8,695 queries in the query log and 3318459, with 6,925 queries in the query log.

We manually labeled the search queries, taking the clicked pages of the query log into consideration. Each query is labeled as being “commercial” or “non-commercial”. In the original data, some query keywords are invalid, e.g. “-”, so we count them as Invalid queries. The distribution of commercial intention in the queries of the Web users we’ve chosen above is shown is the following Table 1.

No	UserID	Commercial	Non-commercial	Invalid
1	42075	295	4868	2
2	117514	412	7029	104
3	2263543	3725	4948	22
4	3318459	758	6072	95

Table 1: OCI distribution of the selected datasets

In the query log data, we first performed preprocessing to remove all invalid queries. We then divided 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.

4.2 Evaluation Metric

For all the experiments in this section, we use standard F1-measure as our evaluation metric, and compare our performance with the baseline method. The standard metrics here include precision, recall, and F1-measure. Precision (P) is defined as the proportion of true positive class members by the system out of all predicted positive class members returned by the system. Recall (R) is the proportion of predicted positive members out of all actual positive class members in the data. F1-measure is the harmonic mean of precision and recall.

In this personalized OCI detection problem. We have:

A: the number of queries correctly classified as having commercial intention.

B: the number of queries classified as having commercial intention.

C: the number of queries that have commercial intention.

So the definition of Precision (P), Recall(R) and F1-Measure is:

$$\text{Precision}(P) = \frac{A}{B}, \text{Recall}(R) = \frac{A}{C}, \text{F1} = \frac{2 \times P \times R}{P + R}$$

4.3 Performance of baseline classifier

For the baseline method, we use the classifier which is now currently available in the Web³, following the work in [5]. We assume the parameter chosen in their Website is the best-tuned so we just compare to this result. The classification result is in the following Table 2.

No	User ID	Precision	Recall	F1-Measure
1	42075	0.843	0.826	0.834
2	117514	0.810	0.836	0.823
3	2263543	0.862	0.819	0.840
4	3318459	0.837	0.825	0.831

Table 2: Baseline Classifier Performance

4.4 Varying the Confidence Parameter θ

We then analyze how different parameters of θ and l will affect our algorithm performance. 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, note that $F1_i$ is the F1-measure calculated for the i^{th} user, as the column No. stated in Table 1.

From Table 3, 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. The reason follows from intuition that when θ is too large, 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. Also it’s noteworthy to mention that 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 won’t help the creation of skip links and the resulting CRF is essentially very similar to a linear-chain CRF.

4.5 Varying Session Length L

In the next experiment, we test whether different parameters of l will lead to large variance in classification accuracy. Ideally, we hope the l we’ve chosen is as close to the real session length as possible, because taking too many query log data into consideration may harm the overall effect, in that similar queries may not belong to the same query session. From the result in the earlier experiment, here we empirically set θ as 0.1. We get the following result in Table 4. From the Table 4, small changes of l won’t change the overall prediction accuracy greatly.

Generally, in almost all of the parameter settings, our personalized OCI detection algorithm **POINT** can substan-

³<http://adlab.msn.com/OCI/OCI.aspx>

tially improve the accuracy of the generalized OCI detection classifier.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented a new algorithm **POINT** to solve the problem of personalized OCI detection. Our work follows from the intuitive motivation that in the same session of a personal query log, semantic similarities and the correlation between nonadjacent queries can help improve the overall classification accuracy. Based on this assumption, we exploited a skip chain CRF model to model the problem as a sequence 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 real query log dataset shows that our algorithm can effectively improve the accuracy of personalized OCI detection.

We plan to carry out several future extensions of this work. In this paper we empirically set the “query session length” as a fixed value. However, it would be more sound if we could automatically determine different lengths that constitute different Web search sessions. Another possible direction is to use the semi-supervised learning framework because it would be too costly to manually label so many items in query logs. Using both labeled and unlabeled data in learning the parameter of skip-chain CRF would be an interesting problem to pursue further research. A third direction is to learn how to use the transfer learning framework to use the queries of other users to improve the accuracy of personalized OCI detection.

Finally, it is also interesting to analyze the effects when we consider take the query logs of similar users into consideration. As mentioned by [12], taking information from related users that have geographical proximity will help Web search. The dataset of our experiment did not allow such an experiment. However, this factor in Web search is also worth considering in query log analysis.

6. REFERENCES

- [1] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. A. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized web query log. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–328, 2004.
- [2] S. M. Beitzel, E. C. Jensen, O. Frieder, D. A. Grossman, D. D. Lewis, A. Chowdhury, and A. Kolcz. Automatic web query classification using labeled and unlabeled training data. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 581–582, 2005.
- [3] A. Z. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.
- [4] P.-A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter. Using odp metadata to personalize search. In *SIGIR*, pages 178–185, 2005.
- [5] H. K. Dai, L. Zhao, Z. Nie, J.-R. Wen, L. Wang, and Y. Li. Detecting online commercial intention (oci). In *Proceedings of the 15th international conference on*

θ	F1 ₁ (Variance)	F1 ₂ (Variance)	F1 ₃ (Variance)	F1 ₄ (Variance)
$\theta = 0.01$	0.874 (0.002)	0.862 (0.003)	0.873 (0.002)	0.831 (0.003)
$\theta = 0.02$	0.885 (0.003)	0.863 (0.003)	0.902 (0.004)	0.873 (0.004)
$\theta = 0.04$	0.883 (0.005)	0.876 (0.004)	0.901 (0.004)	0.926 (0.007)
$\theta = 0.08$	0.902 (0.005)	0.891 (0.003)	0.912 (0.003)	0.910 (0.003)
$\theta = 0.1$	0.927 (0.004)	0.930 (0.004)	0.913 (0.005)	0.915 (0.007)
$\theta = 0.2$	0.911 (0.005)	0.942 (0.005)	0.895 (0.006)	0.901 (0.007)
$\theta = 0.4$	0.891 (0.004)	0.889 (0.007)	0.873 (0.004)	0.897 (0.004)
$\theta = 0.8$	0.842 (0.004)	0.839 (0.004)	0.848 (0.003)	0.834 (0.004)
Baseline	0.834	0.823	0.840	0.831

Table 3: Accuracy with changing parameter θ

L	F1 ₁ (Variance)	F1 ₂ (Variance)	F1 ₃ (Variance)	F1 ₄ (Variance)
$L = 100$	0.919 (0.009)	0.892 (0.011)	0.886 (0.008)	0.896 (0.009)
$L = 200$	0.910 (0.007)	0.902 (0.005)	0.908 (0.009)	0.902 (0.010)
$L = 400$	0.916 (0.006)	0.908 (0.006)	0.915 (0.008)	0.913 (0.009)
$L = 800$	0.920 (0.009)	0.929 (0.008)	0.921 (0.010)	0.924 (0.005)
$L = 1000$	0.927 (0.004)	0.930 (0.004)	0.913 (0.005)	0.915 (0.007)
$L = 2000$	0.928 (0.003)	0.911 (0.004)	0.908 (0.005)	0.916 (0.006)
$L = 3000$	0.919 (0.006)	0.906 (0.002)	0.901 (0.003)	0.899 (0.004)
$L = 4000$	0.915 (0.005)	0.936 (0.003)	0.898 (0.004)	0.898 (0.006)
Baseline	0.834	0.823	0.840	0.831

Table 4: Accuracy with changing parameter L

- World Wide Web*, pages 829–837, 2006.
- [6] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW*, pages 581–590, 2007.
- [7] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW*, pages 517–526, 2002.
- [8] B. J. Jansen. The effect of query complexity on web searching results. *Inf. Res.*, 6(1), 2000.
- [9] I.-H. Kang and G.-C. Kim. Query type classification for web document retrieval. In *SIGIR*, pages 64–71, 2003.
- [10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.
- [11] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *WWW*, pages 391–400, 2005.
- [12] Q. Mei and K. W. Church. Entropy of search logs: how hard is search? with personalization? with backoff? In *Proceedings of the International Conference on Web Search and Web Data Mining*, pages 45–54, 2008.
- [13] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *WWW*, pages 727–736, 2006.
- [14] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, pages 377–386, 2006.
- [15] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang. Query enrichment for web-query classification. *ACM Trans. Inf. Syst.*, 24(3):320–352, 2006.
- [16] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Building bridges for web query classification. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 131–138, 2006.
- [17] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen. Cubesvd: a novel approach to personalized web search. In *WWW*, pages 382–390, 2005.
- [18] C. A. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.
- [19] C. A. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research*, 8:693–723, 2007.
- [20] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 969–976, 2006.
- [21] Q. Zhao, S. C. H. Hoi, T.-Y. Liu, S. S. Bhowmick, M. R. Lyu, and W.-Y. Ma. Time-dependent semantic similarity measure of queries using historical click-through data. In *Proceedings of the 15th international conference on World Wide Web*, pages 543–552, 2006.