Contextual Query Intent Extraction for Paid Search Selection

Pengqi Liu
Microsoft Inc.
Sunnyvale, CA, USA
penliu@microsoft.com

Javad Azimi
Microsoft Inc.
Sunnyvale, CA, USA
jaazimi@microsoft.com

Ruofei Zhang
Microsoft Inc.
Sunnyvale, CA, USA
bzhang@microsoft.com

ABSTRACT

Paid Search algorithms play an important role in online advertising where a set of related ads is returned based on a searched query. The Paid Search algorithms mostly consist of two main steps. First, a given searched query is converted to different sub-queries or similar phrases which preserve the core intent of the query. Second, the generated sub-queries are matched to the ads bidded keywords in the data set, and a set of ads with highest utility measuring relevance to the original query are returned. The focus of this paper is optimizing the first step by proposing a contextual query intent extraction algorithm to generate sub-queries online which preserve the intent of the original query the best. Experimental results over a very large real-world data set demonstrate the superb performance of proposed approach in optimizing both relevance and monetization metrics compared with one of the existing successful algorithms in our system.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Machine Learning**]: [Performance Measures]

General Terms

Algorithm, Design.

Keywords

Paid Search, Query Intent.

1. INTRODUCTION

Paid Search (PS) is one of the largest revenue sources of online advertising companies like Microsoft, where the goal is returning relevant ads for searched queries. Typically, it is preferred to return ads whose Bidded Keywords (BK) are exactly the same as the query. However, it does not happen often as many searched queries are not exactly the same as available BKs in ads data set. To increase ads selection coverage and depth, it is very important to shorten the query by removing tokens which do not have a contribution to the query core intent. For example, the query *buy harry potter dvd online* is almost the same as *harry potter dvd* without losing much user intent. The goal of this work is to drop the tokens that has the least contribution to the query intent.

There are a few related studies in the domain of sentence compression [4] in which the sentences are grammatically sound by assumption. However, in our application, the query is not always grammatically correct which makes the sentence compression methods inapplicable to this problem. Shortening searched queries to return relevant webpages in search engines is another set of algorithms which tackle the similar problem [2], [3]. However, the proposed approaches are not directly applicable for ads world problem.

In this work, we present a novel contextual approach to generate a set of sub-queries online from a given query such that the generated sub-queries preserve the core intent of original query. We first generate all sub-queries with at least 2 tokens from the original query. Next, two sets of feature extraction rules, *Mutual Click Intent* and *Click Intent Rank*, are presented to generate a set of contextual features for each sub-query. Finally, a logistic regression classifier is used to determine the goodness of each sub-query. Experimental results over both online and offline data demonstrate the effectiveness of proposed approach in generating high quality sub-queries comparing to the baseline algorithm.

2. INTENT EXTRACTION ALGORITHM

In this section we introduce our proposed algorithm which consists of three steps, *Sub-queries Generation*, *Feature Extraction* and *Learning Model*.

2.1 Sub-queries Generation

Given a query with n tokens after stopwords removal, all sub-queries with length between 2 and $n-1$ are generated. Then, we extract some features from each sub-query.

2.2 Feature Extraction

In this section, we propose our novel feature extraction algorithms to represent each sub-query as a set of numerical features measuring the intent preserveness of the sub-query.

2.2.1 Mutual Click Intent (MCI)

The MCI score determines the coherence between any pair of tokens in query based on historical PS logs (explaining in section 3). First, given a query q, an undirected graph $G(V, E)$ is constructed where V represents query tokens and E represents the MCI score between any pair of tokens [2]. The MCI score between any token v_i, v_j is calculated as: $MCI_{i,j} = \frac{N^{i,j}_{i,j}}{N^{i,j}_{i,j}+N^{i,j}_{i}+N^{i,j}_{j}+1}$, where $N^{i,j}_{i,j}$ and $N^{i,j}_{k}$ are the

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742740.

number of historical clicked <query, ads BK> pairs that query contains both v_i, v_j and ads BK contain v_i, v_j and v_k correspondingly. As an example, $MCI_{harry,potter} = 0.98$ while $MCI_{cheap,hotel} = 0.52$. Once the graph $G(V, E)$ is constructed, each sub-query q_s is represented as a sub-graph $G_s(V_s, E_s) \in G(V, E)$ where V_s only consists of q_s tokens. Then, for each q_s the following features are calculated: 1) $f_1 = \frac{\sum_{v_i,v_j \in V_s} MCI_{i,j}}{\sum_{v_i,v_j \in V} MCI_{i,j}}$, 2) $f_2 = \frac{\sum_{v_i,v_j \in V_s} MCI_{i,j}/|E_s|}{\sum_{v_i,v_j \in V} MCI_{i,j}/|E|}$, 3) $f_3 = \frac{\max_{v_i,v_j \in V_s} MCI_{i,j}}{\max_{v_i,v_j \in V} MCI_{i,j}}$, 4) $f_4 = \frac{\max_{v_i \in V_s, v_j \in V \setminus V_s} MCI_{i,j}}{\max_{v_i,v_j \in V} MCI_{i,j}}$. Note that, f_4 indicates the max score connecting the two distinct graphs G_s and $G \setminus G_s$, which captures the contextual information of the sub-query q_s in the original query.

2.2.2 Click Intent Rank (CIR)

In this section, we propose a random walk algorithm [3] to quantify the contribution of each token to query intent. First, given a query q with n tokens, a *directed* graph $G(V, E)$ is constructed where V represents query tokens and E represents the transition probability between any pair of tokens. The transition probability from v_i to v_j is calculated offline as: $e_{i,j} = P(v_j|v_i) = \frac{N^{i,j}_{i,j}+1}{N^{i,j}_{i,j}+N^{i,j}_{i}+1}$, which simply means how likely is having the token v_j in BK if the token v_i is already existed in the query and BK given clicked <query, ads BK> logs. Then, PageRank algorithm [1] is applied to this graph to calculate the final CIR for each token as follows:

$$\mathbf{CIR^{t+1}} = \alpha \mathbf{CIR^t E} + (\mathbf{1} - \alpha)\mathbf{U}, \quad (1)$$

where, $\mathbf{CIR^t_{1\times n}}$ is a probability vector over each token at iteration t, $\mathbf{E_{n\times n}}$ is the transition probability matrix between tokens, $\mathbf{U_{1\times n}}$ is a constant vector, and $\alpha \in (0, 1)$ is the damping factor. The CIR_i simply indicates how important token v_i is in the query. As an example, given query *stella artois beer prices*, $CIR_{stella} = 0.35$, $CIR_{artois} = 0.34$, $CIR_{beer} = 0.26$ and $CIR_{prices} = 0.05$. Then, we calculated 6 different features for each sub-query q_s based on CIR: 1) $f_5 = \frac{\sum_{v_i \in V_s} CIR_i}{\sum_{v_j \in V} CIR_j}$, 2) $f_6 = \frac{\sum_{v_i \in V_s} CIR_i/|V_s|}{\sum_{v_j \in V} CIR_j/|V|}$, 3) $f_7 = \frac{\max_{v_i \in V_s} CIR_i}{\max_{v_j \in V} CIR_j}$, 4) $f_8 = \frac{\max_{v_i \in V \setminus V_s} CIR_i}{\max_{v_j \in V} CIR_j}$, 5) $f_9 = \frac{\sum_{v_i \in V \setminus V_s} CIR_i/|V \setminus V_s|}{\sum_{v_j \in V} CIR_j/|V|}$, 6) $f_{10} = \frac{(\prod_{v_i \in V \setminus V_s} CIR_i)^{-|V \setminus V_s|}}{(\prod_{v_j \in V} CIR_j)^{-|V|}}$.

2.3 Learning Model

We train a Logistic Regression (LR) model to calculate the probability of being a good candidate for each sub-query based on its extracted 10 features.

3. EXPERIMENTS

3.1 Data Sets

We used Bing one year's search log in order to automatically generate a training data set which is called D_1. First, all <query, ads BK> pairs which have more than $1k$ impressions are selected. Then, the <query, ads BK> pairs which 1) have Click Through Rate (CTR) more than 20% and 2) all of the ad BK tokens exist in the query, are considered as positive examples. Also, <query, ads BK> pairs which have CTR less than 0.01% are considered as negative examples. AS the result, we have around $20M$ positive pairs and $40M$ negative pairs. The positive pairs are then used to calculate the *MCI* and the transition probability in *CIR*.

We also create a manual data set D_2 by randomly selecting $10k$ <query, ads BK> pairs from searched log in which BK is sub-phrase of query. Some human experts are asked to label them as relevant(positive) or irrelevant(negative) pairs.

3.2 Results

The D_1 data set is splitted into test (20%) and training (80%) data set. The LR parameters are learned using the training data and the proposed approach is evaluated over the test data. We also used D_2 as another test data set. Precision and Recall (for the positive labels) are then used as evaluation metrics. The results are presented in Table 1.

The first two rows show the result of our proposed approach over D_1 and D_2 accordingly. We also compared the proposed approach against *Partial Drop*(PD) algorithm, similar ideas as [3], which drops one or two tokens from a given searched query by considering the CTR impact from historical data. The last two rows in Table 1 present the performance of PD algorithm over both data sets.

The results show that our approach significantly outperforms PD, especially on recall which is more important for PS selection. It also shows a good correlation between both data sets performances. This indicates the effectiveness of proposed mechanism in creating D_1.

Table 1: The proposed approach results.

Methods	Precision	Recall
Proposed Approach (D_1)	0.847%	0.813%
Proposed Approach (D_2)	0.935%	0.576%
Partial Drop (D_1)	0.605%	0.223%
Partial Drop (D_2)	0.931%	0.255%

We have also compared our proposed approach with PD using online real traffic experiment in Bing. The results are presented in Table 2. The second column represents RPM (Revenue per 1000 impressions) and the third column represents CTR. The results show that the proposed algorithm achieves a higher RPM (we use virtual number here for sensitivity reason) comparing to PD without compromising the CTR. Further investigation shows that the RPM improvement mostly comes from increasing the impression rate (recall) which is consistent to the result in Table 1.

Table 2: Online results.

Methods	RPM	CTR
Proposed Approach	701	10.90%
Partial Drop	690	10.93%

4. CONCLUSION AND FUTURE WORK

In this paper, we introduced an approach which generates representative sub-quries from a given searched query. Unlike most of the previous work in PS, the proposed approach utilizes the click intent data and build the model by considering the query context. The experimental results over a very large real world data set and online traffic testing demonstrate the effectiveness of proposed approach in generating more relevant and valuable candidate sub-queries comparing to the baseline algorithm.

5. REFERENCES

[1] The anatomy of a large-scale hypertextual web search engine. *ISDN*, 1998.

[2] G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *SIGIR*, 2009.

[3] K. T. Maxwell and W. B. Croft. Compact query term selection using topically related text. In *SIGIR*, 2013.

[4] E. Pitler and E. Pitler. Methods for sentence compression, 2010.