Intent Boundary Detection in Search Query Logs

Chieh-Jen Wang Kevin Hsin-Yih Lin Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University
Taipei 106, Taiwan
cjwang@nlg.csie.ntu.edu.tw f93141@csie.ntu.edu.tw hhchen@csie.ntu.edu.tw

ABSTRACT
Identifying intent boundary in search query logs is important for learning users' behaviors and applying their experiences. Time-based, query-based, and cluster-based approaches are proposed. Experiments show that the integration of intent clusters and dynamic time model performs the best.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: *Query formulation.*

General Terms
Algorithms, Experimentation, Measurement

Keywords
Intent clustering, intent boundary detection, query log analysis.

1. INTRODUCTION
Search query logs record users' queries along with their clicked URLs. Many researches [1] try to mine common behaviors from logs, and apply the experience to information access. One issue in query log mining is to determine the intent boundaries in a sequence of queries. The static time-based approach considers the average time per intent and employs a time threshold to partition query logs into intent-coherent sessions. The major problem of this approach is that it is easy to leave out information when using a small threshold, and to introduce noise when using a large threshold. The query-based approach computes the average number of queries needed to fulfill an information need, and uses it to recognize intent boundaries. The query-based approach suffers from the same problem. A very small or a very large threshold will result in too little or too much information.

This paper proposes a dynamic time model which considers user comprehension time in querying and clicking URL. First, the model determines a potential intent boundary. Then, intent clusters learned from a Live search query log [2] are adopted to adjust the boundary. For each query, its query terms, query time, and the associated session are recorded in the log. For each click, the clicked URL, the click time, and the associated query are recorded in the log. In total, there are 7,468,628 sessions. Each session may contain one or more intents. To purify the dataset, we propose the following criteria to select 14,242 sessions for intent clustering: 1) session duration is no longer than 60 minutes; 2) at least 3 distinct queries; 3) at least 3 clicked URLs which can be found in the Open Directory Project (ODP).

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
ACM 978-1-60558-896-4/10/07.

2. COMPREHENSION TIME MODEL
A session is a sequence of queries and URLs. The possible transitions between two continuous actions are: $q \rightarrow q'$, $q \rightarrow u$, $u \rightarrow q$ and $u \rightarrow u'$, where q and q' are queries, and u and u' are URLs, respectively. Transition $q \rightarrow q'$ means submitting two queries continuously; transition $q \rightarrow u$ means submitting q and then clicking u reported by this query; transition $u \rightarrow q$ means clicking u reported by one query and then issuing another query q; transition $u \rightarrow u'$ means clicking two URLs reported by one query continuously. $|q \rightarrow q'|$, $|q \rightarrow u|$, $|u \rightarrow q|$ and $|u \rightarrow u'|$ denote the duration time for the corresponding transitions. In addition to the explicit submitting and clicking actions, there are implicit actions in a time interval. For example, clicking u, retrieving u, reading u, deciding to initiate a new query q, and submitting it are actions embedded in transition $u \rightarrow q$.

The duration time $|u \rightarrow q|$ depends on several factors, e.g., Internet bandwidth, users' comprehension ability, and so on. Here we neglect the bandwidth issue, and focus on comprehension. We define the comprehension time h of a specific session as follows, where n is the number of $u \rightarrow u'$ transitions in a session for the user:

$$h = \frac{1}{n}\sum_{i=1}^{n} |u_i \rightarrow u_{i+1}|$$

Given search query logs, we compute all duration time $|q \rightarrow q'|$, $|q \rightarrow u|$, $|u \rightarrow q|$, and $|u \rightarrow u'|$. Then the following average duration time is derived: 1) $|Q \rightarrow Q|$: the average duration time between submitting one query and another one, i.e., 98.75 sec in Live search query logs; 2) $|U \rightarrow U|$: the average duration time between clicking one URL and another one, i.e., 116.32 sec; and 3) $|U \rightarrow Q|$: the average duration time between clicking a URL and then issuing a query, i.e., 236.74 sec.

We define static and dynamic comprehension time models as follows to capture users' comprehension behaviors:

(1) StaticCTime Model: if $|u \rightarrow q| > |U \rightarrow Q|$, we propose a potential boundary before q.
(2) DynamicCTime Model: if $|u \rightarrow q| - h > |U \rightarrow Q| - |U \rightarrow U| + S$, we propose a potential boundary before q, where S is a standard deviation of $U \rightarrow U$ duration time, i.e., 92.39 sec.

3. INTENT CLUSTERING
Intent clustering aims to group sessions of similar intents into a cluster. Query and clicked URLs are clustering features which can be considered. An individual query may be ambiguous in meaning, but queries in the same session can be disambiguated with one another. Clicked URLs are not always relevant to the queries, but this issue is not dealt with in this paper. Similar to a query, a URL may have more than one ODP category

(abbreviated as path hereafter). We resolve the ambiguity by using contextual information. The features of query, URL, path, query+URL, and query+path are explored on complete link and average link clustering algorithms to derive intent clusters.

Given a session of n queries, there may be n possible intent boundaries (defining the last boundary to be right after the last query) and thus n possible segments. We have to compute the similarity of these n segments with all the m intent clusters. The segment of the highest similarity is proposed. We need $m*n$ computations to find boundaries. To reduce time cost, we use time model to propose a potential boundary first. Then we compute the similarity of the segment with all intent clusters, select the cluster with the highest similarity with the segment, and adjust the potential intent boundary (say q_i) using the intent cluster. We try to move the boundary to the right to include one more query q_{i+1} and compute similarity. If the similarity becomes lower, we try to move the boundary to the left to query q_{i-1} and compute the similarity. Otherwise, we try to move the boundary to the right again. The right/left movement procedure is repeated until the similarity becomes lower.

4. EXPERIMENTS AND DISCUSSIONS

We randomly select 1,000 sessions from Live search query log data set [2], and manually label the intent boundaries in each session. Total 1,456 boundaries are identified. Assume ground truth G and system report S are composed of m and n intent boundaries, respectively. Let $\{c_1, c_2, \ldots, c_k\}$ be k common boundaries between G and S. Precision (P), recall (R), and F-Score (F) are defined as follows, where b_i=1 if there does not exist any boundary c in G and S such that c is located between c_{i-1} and c_i, otherwise b_i=0.

$$P = \frac{\sum_{i=1}^{k} b_i}{n} \qquad R = \frac{\sum_{i=1}^{k} b_i}{m} \qquad F = \frac{2 \times P \times R}{P + R}$$

Table 1 shows the performance of four baseline approaches using time or query thresholds. AvgTime which considers a segment spanning 20 minutes as an individual intent is the best baseline. Avg#Queries which regards a segment consisting of 7 queries performs the next. The performance of considering comprehension time, i.e., StaticCTime and DynamicCTime, is behind that of AvgTime and Avg#Queries.

Next we introduce intent cluster to adjust the intent boundary proposed by the DynamicCTime approach. Table 2 shows the results. Intent clusters generated by complete link and average link clustering algorithms on different features are compared. Using URL features is better than using query features. It may be due to that clicked URLs express some sort of users' intents, and users' queries may be ambiguous. Using path feature, denoting ODP category, performs better than using URL features in the complete link algorithm. That meets our expectation because the ODP category is a conceptual representation of URL. Using a Query together with URL/Path is better than using the URL/Path only in these two clustering algorithms. Integrating features of query and path in complete the link algorithm generates the best intent clusters, which are the most helpful to intent boundary identification. The resulting model achieving an F-score 0.6543 is significantly better than the four baselines in Table 1 (Chi-Square test, p-value<0.001).

We also introduce intent cluster to the best baseline, AvgTime. Table 3 shows the results. The tendency is similar to Table 2. Complete link clusters with Query+Path performs better than the AvgTime only approach. However, it still cannot compete with the approach of integrating intent cluster and DynamicCTime. The results reflect again considering individual comprehension time is quite useful for boundary detection.

Table 1. Using Time/Query Constraints

	Precision	Recall	F-Score
AvgTime	0.6368	0.6348	0.6355
Avg#Queries	0.6203	0.6195	0.6196
StaticCTime	0.5171	0.5175	0.5157
DynamicCTime	0.5229	0.5235	0.5217

Table 2. Introducing Intent Clusters to DynamicCTime

CompleteLink	Precision	Recall	F-Score
Query	0.6305	0.6581	0.6441
URL	0.6316	0.6598	0.6452
Path	0.6323	0.6666	0.6492
Query+URL	0.6334	0.6660	0.6495
Query+Path	**0.6409**	**0.6681**	**0.6543**
AverageLink	Precision	Recall	F-Score
Query	0.6292	0.6609	0.6447
URL	0.6341	0.6635	0.6491
Path	0.6324	0.6670	0.6488
Query+URL	0.6358	0.6667	0.6509
Query+Path	**0.6355**	**0.6733**	**0.6539**

Table 3. Introducing Intent Clusters to AvgTime

CompleteLink	Precision	Recall	F-Score
Query	0.6313	0.6284	0.6294
URL	0.6351	0.6319	0.6331
Path	0.6393	0.6370	0.6376
Query+URL	0.6349	0.6325	0.6334
Query+Path	**0.6413**	**0.6384**	**0.6394**
AverageLink	Precision	Recall	F-Score
Query	0.6258	0.6235	0.6242
URL	0.6293	0.6270	0.6278
Path	0.6311	0.6285	0.6293
Query+URL	0.6367	0.6344	0.6352
Query+Path	**0.6382**	**0.6357**	**0.6364**

5. CONCLUSIONS

This paper detects intent boundaries in search query logs. The intent clusters generated by using queries and ODP categories of clicked URLs are proved to be useful.

6. ACKNOWLEDGEMENTS

This work was supported in part by Microsoft Research Asia and by the Excellent Research Projects of National Taiwan University.

7. REFERENCES

[1] Jones, R. and Klinkner, K.L. Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs, In *Proceeding of the 17th ACM CIKM*, 2008, 699-708.

[2] Craswell, N., Jones, R., Dupret, G. and Viegas, E. Workshop on Web Search Click Data, held in conjunction with WSDM 2009.