Computational Linguistics and Chinese Language Processing
Vol. 16, No. 3-4, September/December 2011, pp. 61-76

Intent Shift Detection Using Search Query Logs

Chieh-Jen Wang[*], and Hsin-Hsi Chen[*]

Abstract

Detecting intent shift is fundamental for learning users' behaviors and applying their experiences. In this paper, we propose a search-query-log based system to predict users' intent shifts. We begin with selecting sessions in search query logs for training, extracting features from the selected sessions, and clustering sessions of similar intent. The resulting intent clusters are used to predict intent shift in testing data. The experimental results show that the proposed model achieves an accuracy of 0.5099, which is significantly better than the baselines. Moreover, the miss rate and spurious rate of the model are 0.0954 and 0.0867, respectively.

Keywords: Intent Shift Detection, Intent Analysis, Search Query Logs Analysis.

1. Introduction

Understanding behavior in users' search sessions is important because of the multiple potential applications, such as query recommendation, web page re-ranking, and advertisement arrangement. Several approaches have been proposed to define a session in a sequence of actions between a user and a search engine. For example, the time-based approach employs a time threshold to partition the queries in a fixed time period into a session. Determining a suitable threshold is the major problem of this approach. Information will be lost when a large threshold is adopted. In contrast, noise will be introduced when a small threshold is adopted. The query-based approach postulates that an information need is satisfied with a fixed number of queries. This suffers from a problem similar to the time-based approach. A large or a small threshold will introduce too little or too much information.

Clarifying a boundary in a sequence of queries to form an intent-coherent session is a fundamental task in mining users' behaviors on the World Wide Web. One of the possible approaches to accomplish this task is determining intent shifts in a sequence of queries. An intent shift occurs when the intent of the current query is different from the original search intent during information access. Given a sequence of queries, q_1, q_2, ..., and q_n, there is an intent shift between q_i and q_{i+1} if the intent of q_{i+1} is different from that intent of q_1, q_2, ..., q_i.

[*] Department of Computer Science and Information Engineering, National Taiwan University
E-mail: cjwang@nlg.csie.ntu.edu.tw; hhchen@ntu.edu.tw

Take Figure 1 as an example, which is selected from a real session in a query log dataset. An intent shift occurs at q_4 (the 4th query in the session) because the search intent from q_1 to q_3 is about dogs and the search intent of q_4 is about a gymnasium, which is significantly different from the original search intent.

Query Time	Query	Clicked Time	Clicked URL	
2006-05-23 06:43:17	dog show	2006-05-23 06:43:32	http://www.dogstop.com/shows.htm	
2006-05-23 06:43:45	dogs	2006-05-23 06:43:52	http://dogs.about.com	
2006-05-23 06:44:09	dogs	2006-05-23 06:44:10	http://www.nextdaypets.com	
2006-05-23 07:06:20	cornerstone gym	NULL	NULL	Intent Shift
2006-05-23 07:06:40	cornerstone	NULL	NULL	
2006-05-23 07:06:48	cornerstone mcallen	2006-05-23 07:07:05	http://www.cornerstonefitnessrgv.com/contactus.htm	

Figure 1. An intent shift example.

On the web, users complete their information needs through searching and browsing. They submit queries and click URLs in a session to represent their intents. The interactions between users and search engines are kept in search query logs. As similar search behaviors demonstrate similar intents, such a log dataset is a good resource to investigate common users' behaviors. In this paper, we will mine users' behaviors from search query logs and use them for detecting whether there exist intent shifts or not.

The challenging issue is that there may be more than one search intent in a session. Multiple-intent sessions may have negative effects on the clustering performance and impact later intent shift detection. Detecting intent shifts in a session will result in sessions of better quality and have positive effects on the clustering. This is a chicken and egg problem. In this paper, we propose some strategies to sample sessions in search query logs, disambiguate the ambiguous queries and clicked URLs, extract features from different intent representations, generate intent clusters by different cluster algorithms, and explore different intent shift detection models.

The remainder of this paper is organized as follows. In Section 2, we compare our research with others. Section 3 gives an overview of the proposed system and the major resource used in this work. Section 4 describes the session sampling strategies and an algorithm for query and URL disambiguation. Section 5 describes how to assemble the sessions of similar intent in a cluster. In addition, we apply intent clusters generated by different clustering models to detect intent shifts. Section 6 analyzes the performance of different clustering models and discusses the findings along with their implications. Section 7 concludes the remarks.

2. Related Work

Given a sequence of queries, an intent boundary detection algorithm divides it into several sub-sequences. Each sub-sequence of queries, containing a single information need, forms a session. Jansen, Spink, Blakely, & Koshman (2007) specify that a session is a series of users' interactions toward a single information need. A session is a basic unit for intent clustering because clustering models put together user behaviors of the same intent into a cluster.

Generally speaking, a session is usually identified by hard or soft segmentation. In hard segmentation, a session is segmented by users' actions, such as open/close a browser or login/logout of a search engine, or by some heuristic methods, such as time cutoffs (Silverstein, Henzinger, Marais, & Moricz, 1998; Montgomery & Faloutsos, 2001) or mean session lengths (Silverstein *et al.*, 1998; Jansen, Spink, Blakely, & Koshman, 2007). On the contrary, soft segmentation identifies an intent boundary according to topic shift in query streams (He & Harper, 2002), some category of user intent (Ozmutlu & Cavdur, 2005) or dynamic comprehension time (Wang, Lin, & Chen, 2010). Several algorithms have been proposed for detecting the intent shifts or identifying intent boundaries (Cao *et al.*, 2008).

The task of query classification is to classify the queries into some predefined categories. Queries, however, are usually short and ambiguous. To realize the meanings of queries, researchers have introduced the concept of user intent behind queries (Broder, 2002). Queries were classified by the searcher's intent, such as navigational query, whose immediate intent is to reach a particular web site, informational query, whose intent is to acquire some information, and transactional query, whose intent is to perform some web-mediated activity. Queries are characterized along four general facets, *i.e.*, ambiguity, authority, temporal sensitivity, and spatial sensitivity (Nguyen & Kan, 2007). Manshadi and Li (2009) classify queries into finer categories. Shen *et al.* (2005) employ the Open Directory Project (ODP) taxonomy to represent clicked URLs and investigate the topic transition. Hence, queries have to be disambiguated if we want to know their exact meanings.

3. System Overview

An intent shift detection system is outlined in Figure 2. The MSN Search Query Log excerpt (RFP 2006 dataset) (Craswell, Jones, Dupret, & Viegas, 2009) is the main resource of this work. This data set consists of 14.9 million queries and 12.2 million clicks during a one-month period in May 2006. The MSN Search Query Log excerpt is separated into two files, one named *query* and the other named *click*. *Query* file is described by a set of attributes, including Time, Query, QueryID, and ResultCount, and the *click* file contains attributes like QueryID, Query, Time, URL, and URL Position. Note that these two files are linked through QueryID. In total, there are 7.4 million sessions, which contain the activities of users from the time of the first query submission to the time of a timeout between their web browser and the search engine.

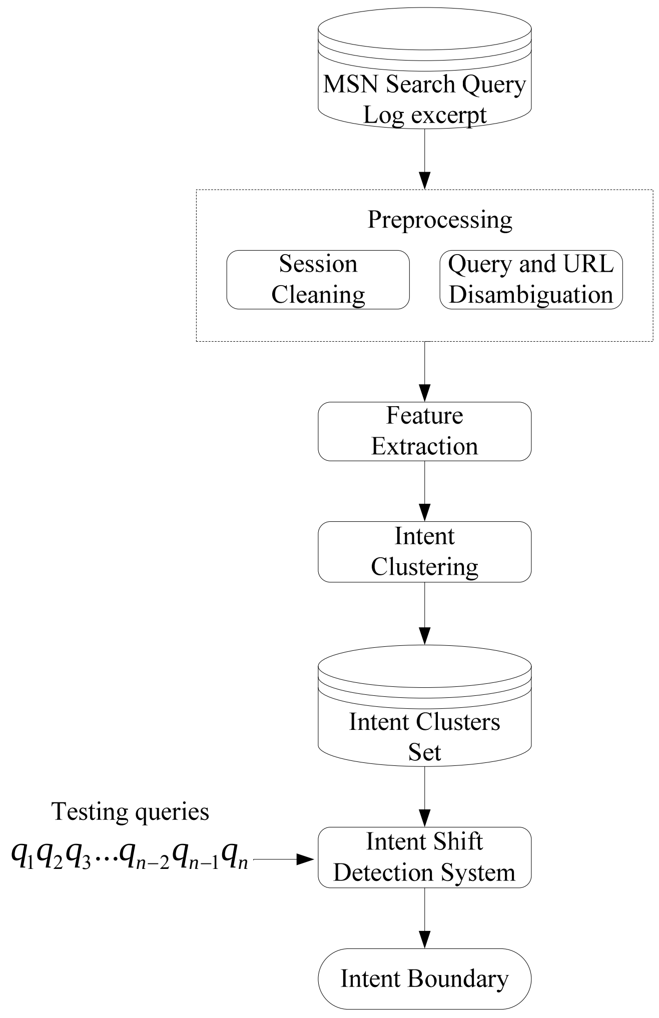
MSN Search Query Log excerpt
Preprocessing
Session Cleaning
Query and URL Disambiguation
Feature Extraction
Intent Clustering
Intent Clusters Set
Testing queries
$q_1 q_2 q_3 \ldots q_{n-2} q_{n-1} q_n$
Intent Shift Detection System
Intent Boundary

Figure 2. A system overview of intent shift detection.

Not all sessions are suitable for constructing the intent shift detection system because of the different backgrounds of users. Besides, a session may contain noise or insufficient information. For these reasons, the search query logs are purified. We take several strategies to get a reliable data set for this study. Queries and URLs are disambiguated based on the ODP (The Open Directory Project, 2002), which is the largest, the most widely distributed human-compiled, and the most comprehensive taxonomy of the websites.

The ODP contains more than 4.5 million websites organized into more than 600 thousand paths. A path (Perugini, 2008) is defined as an ordered hierarchical structure of hyperlink labels from the root category of a directory to a leaf in the ODP. For example, the Academia Sinica website (http://www.sinica.edu.tw/) is assigned a path (Top/Regional/Asia/Taiwan/Education). The root category is "Top" and "Regional" is a sub-category of "Top". The ODP not only contains website annotations edited by volunteers collaboratively, but also provides a natural language description of categories and websites.

After preprocessing, the purified sessions are partitioned into intent clusters by two hierarchical cluster algorithms with queries, clicked URLs, and their corresponding ODP categories. We use the sets of the intent clusters generated by various clustering models to construct the intent shift detection system. To evaluate the performance of the intent shift detection methods on various intent representations, we manually prepare a ground truth. A total of 500 sessions are sampled and annotated for testing. Given a sequence of queries in a testing session, the intent shift detection system will identify whether intent shifts occur.

4. Preprocessing

For creating a reliable dataset, we first clean the MSN Search Query Log excerpt by session cleaning and category disambiguation before intent clustering. Session cleaning filters out potential noise in the MSN Search Query Log excerpt in Section 4.1. The most preferable ODP categories for a clicked URL are selected by a category disambiguation in Section 4.2.

4.1 Session Cleaning

In search query logs, longer sessions containing more queries and clicked URLs tend to contain noise because users have higher probability of changing intents. On the other hand, smaller sessions may not be complete enough to describe the whole information need. In this work, we aim to capture the user intent embedded in sessions as much as we can. Balancing the noiselessness and the completeness is a basic issue for session cleaning.

At the session cleaning stage, we employ the following four filtering strategies.

Filter Strategy #1: Sessions longer than one hour are removed, because a long session may have multiple intents in it. According to the statistics of the MSN Search Query Log excerpt, the longest duration time of a session is more than 99 hours and several intents are observed. Intuitively, it is unlikely for a user to interact constantly with a search engine and maintain only a single intent. As mentioned in related work, several time cutoffs are proposed to segment a session. In specific, Anick (2003) adopts 60 minutes as a time cutoff to segment a session. Therefore, we postulate that the original intent of a user will shift if s/he queries a search engine for more than 60 minutes.

Filter Strategy #2: Users may accomplish their goals with few queries in a small session. For example, a user may submit a navigation query (*e.g*., Academia Sinica), click the official website, and stop the search. A small session does not provide enough information to know his or her exact intent. In this example, the intent may be finding a job or searching for a research institute in Academia Sinica. We consider total number of queries in a session as a filtering strategy. Sessions with less than *n* queries are regarded as small sessions and are removed. As mentioned in related work, a variety of length cutoffs have been proposed by different research projects. Additionally, Silverstein, Henzinger, Marais, & Moricz, (1998) reported the most probable range is between 2 to 3 queries. In our experiments, *n* is set to 3.

Filter Strategy #3: Users may click other search engines during searching and browsing. Nevertheless, how users interact with the search engines is not recorded, so that we have no information about the subsequent actions. As a result, we remove sessions which contain the clicked URLs to other search engines.

Filter Strategy #4: Since our system utilizes ODP categories of URLs for queries and URLs disambiguation, we keep only the sessions where all clicked URLs appear in the ODP.

Table 1 lists the number of remaining sessions after each combination of strategies. After all of the filtering processes, a total of 14,242 sessions remain. The number of the remaining sessions is a low percentage (*e.g*. 0.19%) of the original sessions, but that is not a problem since a huge collection of query logs is available in the real world. More sessions will be generated if more logs are available. Pure sessions are important to generate *correct* results. The basic idea of filtering strategies is avoiding the *garbage in*, *garbage out* problem.

Table 1. Results of session filtering.

Filtering Strategies	Number of Sessions	Remaining Rate (%)
Original	7,468,628	100.00
1	2,815,843	37.70
1 + 2	1,063,906	14.24
1 + 2 + 3	419,383	5.62
1 + 2 + 3 + 4	14,242	0.19

4.2 Category Disambiguation

We postulate that the sequence of clicked URLs in a session is relevant and accomplishes an information need. Therefore, the clicked URLs are co-related and coherent. The clicked URLs and the surrounding URLs are employed to disambiguate the categories of the clicked URL and the associated queries.

For a URL, we consult the ODP to collect all possible paths. As a clicked URL may belong to more than one path, we must find its correct meaning. For example, Brookfield Zoo's website can be mapped to two paths. One specifies that Brookfield Zoo is a travel and tourism location (Top/Regional/North_America/United_States/Illinois/Localities/B/Brookfield/Travel_and_Tourism) and the other one regards it as a biology science location (Top/Science/Biology/Zoology/Zoos_and_Aquariums/North_America/United_States/Illinois). Without disambiguation, it is unclear whether a user clicking on Brookfield Zoo's URL plans a trip to the zoo or is interested in biological science.

The goal is to find the most semantically-coherent path of each URL in a session. Take Figure 3 as an example. A total of 3 URLs, i.e., URL1, URL2, and URL3, have been clicked in a session. URL1, URL2, and URL3 contain 1, 2, and 2 probable paths after consulting the ODP, respectively. There are four possible trails that make up the set of probable paths from each clicked URL. For example, "path11-path21-path31" is one possible trail. The score of a trail is the sum of similarity between a path and the other paths in the session. The similarity of two paths is the number of common categories between these two paths. The number of common categories among paths reflects the degree of intent coherency. Therefore, the trail with the highest score will be selected, and the trail contains the disambiguated ODP categories for the clicked URLs.

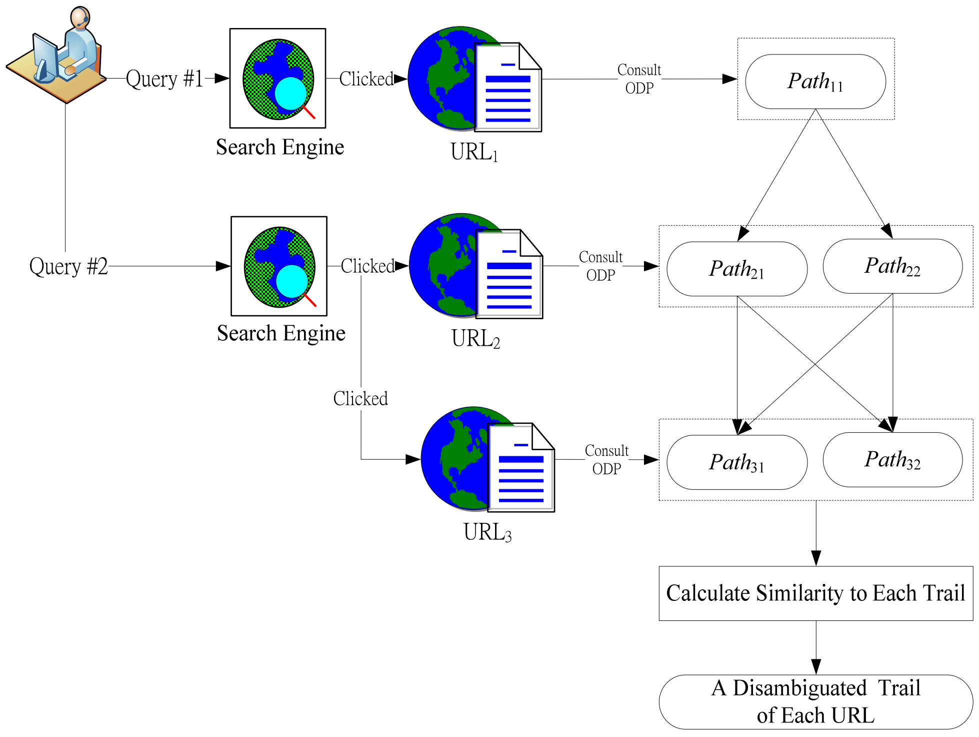

Figure 3. Category disambiguation procedure.

5. Intent Clustering

Sessions are similar if the search intents in them are the same. Sessions of the same intent will be put into a cluster to represent the common behaviors related to the intent. Three sets of features are extracted from each session, and the feature weight is determined by binary and *tf-idf* schemes. Two hierarchical cluster algorithms are used to cluster sessions of similar intent. A clustering model is defined by features and a hierarchical cluster algorithm. A cluster set created by the model is used to detect intent shifts. The performance of the intent shift detection system will be evaluated by three metrics, including *miss rate*, *accuracy*, and *spurious rate*.

5.1 Feature Extraction

Three sets of features, *i.e.*, Query, clicked URLs, and path of clicked URLs with category disambiguation, are considered. The features are used to cluster sessions. Table 2 shows the details of each feature and feature combination. Query terms are the first type of features. In a query feature, query terms are transferred to lower case and stop words are removed, but not stemmed. In addition, a bag-of-words strategy is employed so that two queries consisting of identical terms in any orders are regarded as the same. A complete URL is the second type of features. A disambiguated path is the third types of features. That is, only the best ODP categories are selected as features in this type. Different combinations of the above 3 types of features are shown from the 4th -6th rows.

Table 2. Description of each feature set.

Feature Name	Description
Query	query terms as features
URL	URLs as features
Path	disambiguated ODP categories of URLs as features
Query+URL	query terms and URLs as features
Query+Path	query terms and disambiguated ODP categories as features
Query+URL+Path	query terms, URLs and disambiguated ODP categories as features

The weight of a feature is determined by two possible schemes: binary or tf-idf (Salton & Buckley, 1988). In the binary setting, the weight of a feature is set to 1 if the feature appears in the session, 0 otherwise. In the tf-idf setting, the weight of a feature is defined by Equation (1):

$$w_{i,s} = (0.5 + \frac{0.5\, freq_{i,s}}{\max_s \, freq}) \times log \frac{N}{n_i} \tag{1}$$

where $\max_s freq$ is the maximum feature frequency in session s, $freq_{i,s}$ is the frequency of feature i in session s, N is the total number of sessions, and n_i is the number of sessions where feature i appears.

5.2 Clustering Models

After preprocessing, there were 14,242 sessions for intent clustering. We randomly sampled 500 sessions used for the later evaluation. The remaining 13,742 sessions were clustered by the average link and the complete link hierarchical clustering algorithms with different sets of features. The similarity between two sessions was determined by Euclidean distance. In the experimental setup, there are a total of 24 clustering models since the combinations of 2 clustering hierarchical algorithms, 2 feature weight schemes, and 6 features form the clustering models.

The two hierarchical clustering algorithms need a similarity threshold to separate an agglomerative hierarchical cluster tree into clusters. The similarity between two sessions in the agglomerative hierarchical cluster tree is represented by Euclidean distance. Different intents may be put into a cluster when a loose similarity threshold is selected. On the other hand, a cluster may be divided into more than one smaller cluster when a tight similarity threshold is adopted. The tighter similarity threshold is selected in the experiment due to the fact that the resemblance of sessions is improved in an intent cluster. For all clustering models, setting the threshold to 1 produces more than one cluster and setting the threshold to 2 produces a cluster. We use Algorithm 1 to select the largest threshold between 1 and 2. A hierarchical cluster tree is separated into n clusters by a threshold s (Line 4). The search procedure is stopped if increasing a value of similarity threshold is accurate to 5 decimal points. Each clustering model constructs an intent cluster set by a similarity threshold. The

Algorithm 1. Searching a similarity threshold

```
Input: An agglomerative hierarchical cluster tree $t$
Output: A similarity threshold $s$
1: $s \leftarrow 1$ and $d \leftarrow 0$
2: while ($d<5$)
3:         $d \leftarrow d + 1$
4:         $n \leftarrow$ Cluster($t$, $s$)
5:         $i \leftarrow 1$
6:              While ($i \leq 9$)
7:                   $u = s + i \times 10^{-d}$
8:                   if ($n \neq$ Cluster($t$, $u$)) then $s \leftarrow u$
9:                   end if
10:                  $i \leftarrow i + 1$
11:             end While
12: end while
13: return $s$ as the similarity threshold
```

number of clusters by the 24 clustering models ranges from 3,960 to 4,756, and the details are shown in Table 3.

Table 3. The number of intent clusters created by different clustering models.

Feature	Binary		*tf-idf*	
	Average Link	Complete Link	Average Link	Complete Link
Query	4612	4518	4756	4688
URL	4555	4455	4655	4566
Path	4481	4369	4578	4489
Query+URL	4352	4288	4429	4399
Query+Path	4222	4186	4295	4258
Query+URL+Path	4163	3960	4200	4145

5.3 Intent Shift Detection

Note that the goal of this work is to predict the intent shift. Given a query sequence, $q_1q_2...q_n$, an intent shift at q_j is defined as an intent switch between q_j and q_{j+1}. Algorithm 2 is an intent cluster based intent shift detection algorithm. Given a testing query sequence, q_1, q_2, …, q_n, an approximate strategy considers a span of d queries to form a potential segment. The d value in Algorithm 2 is set to 5, which is the average length of the 13,742 sessions. In this way, a rough segment, q_1, q_2, …, q_5, is derived.

Then, the following procedure finds the most similar intent cluster c^* to identify the intent boundary. The segment, q_1, q_2, …, q_5, is represented by queries, clicked URLs, and disambiguated ODP categories. Category descriptions, URL snippets in ODP, and anchor texts of the clicked URLs are also used if they exist. We measure the similarity of the segment with all intent clusters, and assign the intent cluster with the highest similarity to c^*. LuceneSim (Line 1) is adopted as the similarity function, which is derived from the vector space model[1]. A document whose vector is closer to the query vector gets a higher similarity score. The similarity function is shown as follows:

$$S(Q,D)=\frac{|Q\cap D|}{|Q|}\times\frac{1}{\sum_{t\in Q}(1+log\frac{N}{df(t)+1})^2}\times\sum_{t\in Q\cap D}\frac{\sqrt{c(t,D)}\times(1+log\frac{N}{df(t)+1})^2}{\sqrt{|D|}}$$

where $|Q \cap D|$ is the number of terms that occur in both query Q and document D, $|Q|$ is the length of query Q, $df(t)$ is the number of documents that contain term t, $|D|$ is the length of document D, N is the number of documents in the collection, and $c(t,D)$ is the number of

[1] http://lucene.apache.org/java/docs/.

occurrences of term *t* in document *D*.

We move the boundary from the first query q_1 and compute the similarity between the query vector represented by q_1 and the *c**. Then, we move the boundary to the right to include one more query q_2 and compute the similarity between the query vector represented by the enlarged segment (q_1 and q_2) and *c**. If the similarity becomes lower, q_2 does not belong to the same intent. The right movement procedure is repeated until the similarity becomes lower or the final query is considered. The main idea of the algorithm is: the similarity scores are calculated by a selected intent cluster and a query vector that is represented by queries, clicked URLs, and disambiguated ODP categories. If the intent of the enlarged query is different from the original query vector, the newly-formed query vector contains information different from the selected intent cluster, so the similarity becomes lower.

Algorithm 2. Detecting intent shift

```
Input: A query sequence $q_1q_2...q_n$, an intent cluster set H
Output: an intent shift at $q_{i-1}$
 1:  $c^* \leftarrow \text{argmax}_{c \in H}$ LuceneSim($c$, $q_1q_2...q_d$)
 2:  $g \leftarrow 0$ and $i \leftarrow 1$
 3:  while ($i \le n$)
 4:      if (LuceneSim($c^*$, $q_1q_2...q_i$) $\ge g$)
 5:         $g \leftarrow$ LuceneSim($c^*$, $q_1q_2...q_i$)
 6:      else
 7:         break while
 8:      end if
 9:      $i \leftarrow i + 1$
10:  end while
```

5.4 Evaluation

For evaluating the performance of intent shift detection, a test dataset is constructed as follows. A group of annotators assess the 500 testing sessions according to the intent purity and type. Of the 500 sessions, 355 sessions are annotated single intent, which means all of the queries and clicked URLs belong to the same intent. The intents of the other 145 sessions are tagged ambiguously. This means the intent type cannot be recognized in these sessions. We append each pair of single intent sessions, forming $355^2 = 126{,}025$ new testing query sequences. If two sessions in a pair contain different intents, the correct intent shift location is at the last query of the first session. If two sessions in a pair contain the same intent, the correct intent shift location is at the last query of the second session. Consider an example. Given two sessions, the first session contains *m* queries ($q_{1,1}$, $q_{1,2}$, …, $q_{1,m}$) and the second session contains *n* queries ($q_{2,1}$, $q_{2,2}$, …, $q_{2,n}$). Therefore, the test query sequence includes *m*+*n* queries

($q_{1,1}$, $q_{1,2}$, …, $q_{1,m}$, $q_{2,1}$, $q_{2,2}$, …, $q_{2,n}$). If the two sessions contain the same intent, then a correct intent shift location exists at $q_{2,n}$, otherwise at $q_{1,m}$.

As previously mentioned, a total of 24 intent cluster sets were created by different clustering models. Each of the 24 cluster sets was input into Algorithm 2 to predict intent shifts in the 126,025 testing query sequences.

We evaluate a testing query sequence according to the following evaluation metrics. Equations (2) and (3) define *miss distance* and *spurious distance* between the predicted intent shift location q_{sp} of the system and the intent shift location q_{gt} of ground truth, where *sp* and *gt* are integers and denote the location of a testing query sequence. If $sp < gt$, *i.e.*, the system-predicted intent shift is shorter than ground truth, we calculate a miss distance by Equation (2). In contrast, $sp > gt$ means the system-predicted intent shift is larger than ground truth. In this case, we calculate a spurious distance by Equation (3). When $sp = gt$, *i.e.*, the system prediction and ground truth is exactly matched, we compute the accuracy.

$$miss\ distance = \frac{gt - sp}{gt} \tag{2}$$

$$spurious\ distance = \frac{sp - gy}{gt} \tag{3}$$

Table 4 shows the performance of three baselines using query cutoffs or topic shift. 3QueryCutoff, proposed by Silverstein, Henzinger, Marais, & Moricz (1998), considers a segment consisting of 3 queries. Avg#Queries regards 5 queries as a segment, which shows the average number of queries in sessions of the MSN Search Query Log excerpt. TopicShift is an algorithm proposed by He, Göker, & Harper (2002) to detect intent shifts. They proposed eight search patterns to formulate the topic transformations and calculate a probability of topic transformation to detect intent shifts.

The *Miss Rate*, *Accuracy,* and *Spurious Rate* are computed as Equations (4), (5), and (6), respectively. Decreasing *Miss Rate* and *Spurious Rate* and increasing *Accuracy* are the goal of our prediction system. TopicShift achieves the best performance on *Accuracy* among the three baselines. The *Miss Rate*, *Accuracy,* and *Spurious Rate* of TopicShift are 0.2676, 0.3087, and 0.0956, respectively.

$$miss\ rate = \frac{\sum_{i=1}^{n} miss\ distance}{n} \tag{4}$$

$$spurious\ rate = \frac{\sum_{i=1}^{n} spurious\ rate}{n} \tag{5}$$

$$accuracy = \frac{\#of\ \ testing\ data\ sp = gt}{n} \tag{6}$$

Table 4. Performance of three baselines.

	3QueryCutoff	Avg#Queries	TopicShift
Miss Rate	0.2722	0.2329	0.2676
Accuracy	0.2999	0.1920	0.3087
Spurious Rate	0.0148	0.0190	0.0956

Table 5 and Table 6 show Miss Rate, Accuracy, and Spurious Rate of introducing intent clusters created by the average link clustering algorithms with binary and tf-idf feature weight schemes, respectively. Intent clusters generated by different clustering models are compared. Feature weight with binary is better than tf-idf. Using URL features is better than using Query features. This may be due to the fact that clicked URLs express clearer user intents and that users' queries may be ambiguous. Using Path feature performs better than using URL features. This meets our expectation because an ODP category is a conceptual representation of a URL. Using a Query together with URL/Path is better than using the URL/Path only. Using a Query together with URL+Path is better than using a Query together with URL/Path.

We also introduce intent clusters created by the complete link clustering algorithms with binary and tf-idf feature weight schemes, respectively. Table 7 and Table 8 show the results. The tendency is similar to Table 5 and Table 6. The intent clusters created by the clustering model integrating the features of Query, URL and Path in the complete link clustering algorithm perform the best on intent shift detection. The clustering model achieving accuracy 0.5099 is significantly better than the baselines in Table 4 (p-value<0.001). The results show that considering intent clusters are quite useful for intent shift detection.

Table 5. Introducing the intent clusters created by the average link clustering algorithm with binary weight scheme

Binary	Query	URL	Path	Query + URL	Query + Path	Query + URL + Path
Miss Rate	0.1711	0.1468	0.1493	0.1346	0.1362	0.1118
Accuracy	0.4122	0.4455	0.4602	0.4651	0.4752	0.4866
Spurious Rate	0.0934	0.0686	0.0720	0.0730	0.0751	0.0791

Table 6. Introducing the intent clusters created by the average link clustering algorithm with tf-idf weight scheme

tf-idf	Query	URL	Path	Query + URL	Query + Path	Query + URL + Path
Miss Rate	0.1539	0.1406	0.1317	0.1473	0.1279	0.1206
Accuracy	0.3834	0.4316	0.4386	0.4419	0.4579	0.4655
Spurious Rate	0.1258	0.0846	0.0879	0.0790	0.0990	0.0891

Table 7. Introducing the intent clusters created by the complete link clustering algorithm with binary weight scheme

Binary	Query	URL	Path	Query + URL	Query + Path	Query + URL + Path
Miss Rate	0.1889	0.1548	0.1469	0.1361	0.1306	0.0954
Accuracy	0.4142	0.4548	0.4629	0.4688	0.4795	0.5099
Spurious Rate	0.0984	0.0703	0.0742	0.0815	0.0763	0.0867

Table 8. Introducing the intent clusters created by the complete link clustering algorithm with tf-idf weight scheme

tf-idf	Query	URL	Path	Query + URL	Query + Path	Query + URL + Path
Miss Rate	0.1529	0.1536	0.1469	0.1431	0.1440	0.1359
Accuracy	0.4042	0.4395	0.4490	0.4559	0.4601	0.4690
Spurious Rate	0.1243	0.0662	0.0750	0.0728	0.0730	0.0699

6. Discussion

We measured the performance of the intent cluster sets produced by the 24 intent clustering models. The performance of a cluster set was determined by judging whether each cluster in the set has a coherent intent. Our evaluation strategy was to devise a cluster based intent shift detection system that utilizes an intent cluster set to perform intent shift detection. The model's accuracy depends on the intent coherency of the clusters. Hence, the cluster set that achieves the highest performance of intent shift detection has the most intent-coherent clusters.

Tables 5-7 show a common phenomenon: Miss Rate is higher than Spurious Rate. This shows that our system is conservative for enlarging the intent boundary. In Algorithm 2, the intent shift query is identified if the current similarity is lower than the previous one. Therefore, we may introduce a relaxation strategy that is determined by the similarity score multiplying a weight. This means the drop must be sharp enough to have at least a fixed percentage of the previous similarity. Along the way, the strategy will decrease Miss Rate but Spurious Rate may be increased. It trades off Miss Rate and Spurious Rate.

7. Conclusion and Future Work

This paper predicts intent shifts in the MSN Search Query Log excerpt. The intent clusters generated by using Query, URL, and Path are proven to be useful for this work. We evaluated the intent clusters created by different intent clustering models from the aspects of *Miss Rate*, *Accuracy*, and *Spurious Rate*. The experimental results show that the complete link cluster algorithm is better than the average link cluster algorithm in almost all evaluation metrics.

We will explore the uses of the intention clusters learned from search query logs in one language (e.g., MSN Search Query Log excerpt) to identify the intent shifts of query logs in another language (e.g., SogouQ Query Log), and we will compare the ways of expressing intent in different languages, different areas, and different cultures.

References

Anick, P. (2003). Using terminological feedback for web search refinement: a log-based study. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval,* 88-95. Toronto, Canada: ACM.

Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, 36(2), 3-10.

Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., & Li, H. (2008). Context-aware query suggestion by mining click-through and session data. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining,* 875-883. Las Vegas, Nevada, USA.

Craswell, N., Jones, R., Dupret, G., & Viegas, E. (2009). Proceedings of the 2009 workshop on Web Search Click Data, 95. Barcelona, Spain.

He, D., Göker, A., & Harper, D. J. (2002). Combining evidence for automatic web session identification. Inf. *Process. Manage.*, 38(5), 727-742.

He, D., & Harper, D. J. (2002). Combining evidence for automatic Web session identification. INFORMATION *PROCESSING AND MANAGEMENT*, 38, 727-742.

Jansen, B. J., Spink, A., Blakely, C., & Koshman, S. (2007). Defining a session on Web search engines: Research Articles. *Journal of the American Society for Information Science and Technology*, 58(6), 862-871.

Manshadi, M., & Li, X. (2009). Semantic tagging of web search queries. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2,* 861-869. Suntec, Singapore.

Montgomery, A. L., & Faloutsos, C. (2001). Identifying Web Browsing Trends and Patterns. *Computer*, 34(7), 94-95.

Nguyen, V., & Kan, M. (2007). Functional Faceted Web Query Analysis. In *Query Log Analysis: Social And Technological Challenges. A workshop at the 16th International World Wide Web Conference (WWW 2007).*

Ozmutlu, H., & Cavdur, F. (2005). Application of automatic topic identification on Excite Web search engine data logs. *Information Processing & Management*, 41(5), 1243-1262.

Perugini, S. (2008). Symbolic links in the Open Directory Project. *Information Processing and Management: an International Journal*, 44(2), 910-930.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5), 513-523.

Shen, X., Dumais, S., & Horvitz, E. (2005). Analysis of topic dynamics in web search. In *Special interest tracks and posters of the 14th international conference on World Wide Web,* 1102-1103. Chiba, Japan.

Silverstein, C., Henzinger, M., Marais, H., & Moricz, M. (1998). Analysis of a Very Large AltaVista Query Log. *Technical Note: Digital Equipment Corporation*.

The Open Directory Project. (2002). About the Open Directory Project. Retrieved October 26, 2010, from http://www.dmoz.org/about.html

Wang, C., Lin, K. H., & Chen, H. (2010). Intent boundary detection in search query logs. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval,* 749-750. Geneva, Switzerland.