Inf Retrieval J (2015) 18:504–529
DOI 10.1007/s10791-015-9271-1

Mining and ranking users' intents behind queries

Pengjie Ren[1] · Zhumin Chen[1] · Jun Ma[1] · Shuaiqiang Wang[2] · Zhiwei Zhang[3] · Zhaochun Ren[4]

Received: 19 May 2015 / Accepted: 28 September 2015 / Published online: 9 October 2015

Abstract How to understand intents behind user queries is crucial towards improving the performance of Web search systems. NTCIR-11 IMine task focuses on this problem. In this paper, we address the NTCIR-11 IMine task with two phases referred to as *Query Intent Mining* (*QIM*) and *Query Intent Ranking* (*QIR*). (I) *QIM* is intended to mine users' potential intents by clustering short text fragments related to the given query. (II) *QIR* focuses on ranking those mined intents in a proper way. Two challenges exist in handling these tasks. (II) How to precisely estimate the intent similarity between user queries which only consist of a few words. (2) How to properly rank intents in terms of multiple factors, e.g. relevance, diversity, intent drift and so on. For the first challenge, we first investigate two interesting phenomena by analyzing query logs and document datasets, namely "*Same-Intent-Co-Click*" (*SICC*) and "*Same-Intent-Similar-Rank*" (*SISR*). *SICC* means that when users issue different queries, these queries represent the same intent if they click on the same URL. *SISR* means that if two queries denote the same intent, we should get similar search results when issuing them to a search engine. Then, we propose similarity functions for *QIM* based on the two phenomena. For the second challenge, we propose a novel intent ranking model which considers multiple factors as a whole. We perform extensive experiments and an interesting case study on the Chinese dataset of NTCIR-11 IMine task. Experimental results demonstrate the effectiveness of our proposed approaches in terms of both *QIM* and *QIR*.

✉ Zhumin Chen
chenzhumin@sdu.edu.cn

Pengjie Ren
jay.ren@outlook.com

[1] Department of Computer Science and Technology, Shandong University, Jinan 250101, China

[2] Department of Computer Science and Information Systems, Jyväskylä University, 40100 Jyväskylä, Finland

[3] Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

[4] ISLA, Amsterdam University, 1098 XH Amsterdam, The Netherlands

Keywords Information retrieval · Query understanding · Intent mining · Intent ranking · NTCIR-11 IMine task

1 Introduction

Most queries are short, ambiguous and multifaceted. For example, the query "jaguar" may refer to the animal, the car or the software. Actually people frequently issue queries with more than one unspecified intent (Hu et al. 2012; Sakai et al. 2013; Qian et al. 2013). Besides, users' information needs are multifaceted, exploratory, and a same query may imply different information needs of users. For the query "swine flu", doctors may be interested in the pathogenesis and treatment solutions, while patients may look for the transmissions and preventive measures.

Recently understanding the intents behind users' queries has attracted significant attention in information retrieval (IR) (Dou et al. 2011a; Dang and Croft 2012, 2013). It is one of the most fundamental problems in IR systems. For desktop users, it can help to diversify the search results to cover as many intents as possible in the first page in order to satisfy most users' information needs. For mobile users, it can help to infer a user's exact intent to personalize the search results in the limited screen (Beg and Ahmad 2007; Jiang and Tan 2009). Besides, it is also a crucial issue in many IR problems such as query suggestion and expansion (Hu et al. 2012; Qian et al. 2013; Liao et al. 2011; Biancalana et al. 2013).

In this study, we focus on the NTCIR-11 IMine task. Our efforts are made in two phases: (I) *Query Intent Mining* (*QIM*) and (II) *Query Intent Ranking* (*QIR*). For a given query, *QIM* tries to discover a two-level intent hierarchy with a clustering process from candidate text fragments[1] of the query. *QIR* tries to rank each layer of the two-level intent hierarchy in terms of relevance, diversity and intent drift.

In particular, in Phase (I), we first investigate two interesting phenomena by analyzing query logs and document datasets, namely "*Same-Intent-Co-Click*" (*SICC*) and "*Same-Intent-Similar-Rank*" (*SISR*). *SICC* means that when users issue different queries, these queries may represent the same intent if they click on the same URL. *SISR* means that if two strings denote the same intent, we should get similar search results when issuing them to a search engine. Then we define similarity functions based on the two phenomena as well as syntactic and semantic characteristics. After that, we obtain an integrated similarity function with a supervised learning approach. Finally, we propose an affinity propagation-based clustering algorithm (Frey and Dueck 2007) for discovery of the intents and construction of the two-level intent hierarchy to interpret a query. In Phase (II), we propose ranking models for each layer of the two-level intent hierarchy by considering multiple factors, e.g. relevance, diversity, novelty, time. We further prove that our ranking model is a monotone and submodular function, and thus we can get an approximation of $(1 - \frac{1}{e})$ with a greedy algorithm (ALGORITHM 2) (Cornuejols et al. 1977; Nemhauser et al. 1978).

[1] In this paper, we use the dataset from NTCIR-11 IMine. IMine offers query suggestions, related queries and so on (from multiple commercial search engines) as the candidate text fragments for the query whose intents are to be mined. But technically, a candidate text fragment can be any text fragment related to the query.

We perform extensive experiments and an interesting case study on the dataset of NTCIR-11 IMine task.[2] Experimental results demonstrate the effectiveness of our proposed approaches in terms of both intent mining and ranking.

To sum up, our main contributions are listed as follows.

- We investigate two significant phenomena by analyzing user behavior data in depth, based on which effective similarity functions are proposed to mine a two-level intent hierarchy.
- We propose a supervised learning approach rather than traditional heuristic approaches to integrate similarity functions into a general function for clustering.
- We present ranking models to order the two-level intent hierarchy considering multiple factors, including relevance, diversity and time, etc.
- We prove that our ranking model is a monotone and submodular function, and thus we can get an approximation of $(1 - \frac{1}{e})$ with a greedy algorithm.

The rest of this paper is organized as follows. Section 2 surveys the related work. Section 3 formulates the intent mining and intent ranking problems. Sections 4 and 5 present the clustering-based intent mining approach and intent ranking models respectively. Section 6 reports the experimental results and a case study. Section 7 concludes the paper and discusses our future work.

2 Related work

2.1 Query intent mining

Existing literature about query intent mining can be classified into two categories: *Clustering/Classifying Search Results or URLs* and *Clustering/Classifying Query Related Terms*. We will discuss them respectively.

Clustering/Classifying Search Results or URLs.

Given a query, this category of approaches consider each intent behind the query as a set of search results or URLs, which can be straightforwardly discovered with clustering or classification algorithms.

Chen and Dumais classified search results into predefined hierarchical categories such as Yahoo! or LookSmart's Web directory (Chen and Dumais 2000). Wen et al. clustered similar queries according to their contents and user logs. They proposed a similarity function based on query and search results content to compare two queries (Wen et al. 2001). Carlos Cobos et al. introduced a new description-centric algorithm for the clustering of web results, called WDC-CSK, which is based on the cuckoo search meta-heuristic algorithm, k-means algorithm, Balanced Bayesian Information Criterion, split and merge methods on clusters, and frequent phrases approach for cluster labeling (Cobos et al. 2014). Zeng et al. (2004) clustered search results into different groups with highly readable names. Given a query and the ranked list of titles and snippets, they first extracted salient phrases as candidate cluster names with a regression model learned from human labeled training data. Then, the documents were assigned to relevant salient phrases to form candidate clusters. Wang and Zhai first learned a given query's aspects from users' query

[2] http://www.thuir.org/IMine/.

logs with the star clustering algorithm. Then they categorized and organized the search results according to the learned aspects (Wang and Zhai 2007).

Beeferman and Berger first built a bipartite graph with the click through data consisting of user queries and clicked URLs. Then, they applied an agglomerative clustering to the graph to group related queries and URLs (Beeferman and Berger 2000). Hu et al. (2012) first found two interesting phenomena from user behavior: One Subtopic Per Search and Subtopic Clarification By Keyword. The former means that if a user clicks multiple URLs in one query, then the clicked URLs tend to represent the same facet. The latter means that users often add additional keywords to expand the query in order to clarify their search intent. Based on the two phenomena, they clustered all the clicked URLs and corresponding queries, where each cluster represents an intent. Qian et al. (2013) first classified query intents into two types according to their variation over time line: constant and bursty. Then they regarded query logs as a constantly data stream and divided it into variable-length partitions. Finally, they clustered each partition into different groups of URLs which represent intents. Cao et al. (2008) summarized similar queries into concepts by clustering the click-through bipartite of queries and URLs recorded from query logs. Fujita et al. (2010) used a random walk approach on query-URL bipartite graph to discover facet attributes of queries. Radlinski et al. (2010) discovered intents of queries using query logs. For a given query, they first identified a set of possibly related queries, and then used the random walk similarities algorithm to find intent clusters. Sadikov et al. (2010) clustered the refinements of user queries to mine underlying user intents. They modeled user behavior as a Markov graph combining document click and session co-occurrence information. And then they performed multiple random walks on the graph to get clusters.

Clustering/Classifying Query Related Terms.

Existing work belonging to this direction considers the intent of a query as a set of candidate sub-intents, i.e. query related terms (Xue et al. 2011). These candidate sub-intents may come from many sources such as query suggestions from search engines, related search queries from user query logs and so on. Currently, this domain is one of the hot topics in query intent mining (Aiello et al. 2011).

Moreno et al. (2014) proposed an algorithm called Dual C-Means to cluster search results in dual-representation spaces with query logs. Radlinski et al. (2010) first found similar queries as candidates for a given query from query logs. Then they used a click-through bipartite graph to refine these similar queries. Finally they grouped the candidates into the same clusters. Dang et al. (2011) generated reformulations which represent possible intents of a query by clustering reformulated queries generated from publicly available resources as candidates, e.g., anchor text. Wang et al. (2013) suggested using surrounding text of query terms in top retrieved documents to mine intents and rank them. They first extracted text fragments containing query terms from different parts of documents as candidates. Then they grouped similar candidates into clusters and generate a readable string for each cluster. Roy et al. (2014) took a deeper look at query intent, zooming in on individual words as possible indicators of user intent. They provided a taxonomy of intent words derived through rigorous manual analysis of large query logs.

Recently, this problem has been stressed by NTCIR,[3] which has a task for intent mining and ranking [INTENT-1 of NTCIR-9,[4] INTENT-2 of NTCIR-10,[5] IMine of NRCIR-11

[3] http://research.nii.ac.jp/ntcir/index-en.html.

[4] http://www.thuir.org/intent/ntcir9/.

[5] http://research.microsoft.com/en-us/people/tesakai/intent2.aspx.

(see footnote 2)]. The task consists of two phases: intent mining and ranking. Participants of the task propose many methods (Sakai et al. 2013). Xue et al. (2011) proposed a method which achieves the best performance in English data in terms of *D#-nDCG*. They first extracted candidate sub-intents from query recommendation, Wikipedia and the click through data. Then they evaluated the similarity of every two sub-intents based on the click through data. If the similarity is larger than a predefined threshold, the two sub-intents are judged to belong to the same intent. Yu and Ren (2013) presented a method which gets the best performance in Chinese data in terms of *D#-nDCG*. Specifically, they first classified intents into two types: role-explicit topic and role-implicit topic. For the role-explicit topics, they constructed a modifier graph based on the set of co-kernel-object strings. Then, the modifier graph was decomposed into clusters with strong intra-cluster interaction and relatively weak inter-cluster interaction. Each modifier cluster intuitively reveals a possible intent. For the role-implicit topics which generally express single information needs, they directly used the extracted sub-intents as intents.

2.2 Intent ranking

Related work on intent ranking coexists with intent mining and is mainly included in INTENT/IMine task of NTCIR (see footnote 3).

Wang et al. ranked queries' sub-intents by optimizing both their relevance and diversity. They first estimated a relevance score for a sub-intent considering three aspects: (1) the relevance of sub-intents to the given query; (2) the importance of sub-intents, which partially reflects popularity; and (3) the readability of sub-intents. Then they ranked sub-intents by balancing relevance and diversity. They tried to achieve the goal that major sub-intents were ranked higher and top sub-intents could cover as many different intents as possible (Wang et al. 2013). Xue et al. (2011) extracted and ranked sub-intents at NTCIR-9 INTENT-1 Task. When sub-intents were extracted from the query recommendation of search engines, they ranked them according to the number of search engines which recommend the sub-intents. When sub-intents were selected from the bipartite graph of query logs, they ranked them according to the number of common clicked URLs when user search the query and its sub-intents. Finally they re-ranked these sub-intents by their term frequencies in the clicked titles and snippets. Yu and Ren (2013) classified intents into two types: role-explicit topic and role-implicit topic. For the ranking of sub-intents in role-explicit topic, they defined a quality function of the list with top-*k* sub-intents which consists of three parts: popularity of topic, probability distribution of modifiers, and effectiveness of a sub-intent string expressing a sub-intent. For the ranking of sub-intents in role-implicit topics, they directly generated the ranked list through semantic similarities leveraging on lexical ontologies.

In summary, although there is a growth in researches investigating users' intents of queries recently, there are still some issues to resolve. Firstly, most of current similarity metrics for sub-intents are usually constructed based on a single perspective, i.e. either from query logs only or from documents collections only. In addition, the combination of different similarity functions from multiple resources are usually defined heuristically. This can not precisely estimate the similarity between sub-intents because of their short text characteristics. Secondly, existing approaches consider query intent mining and ranking from a static viewpoint. They ignore the issue of intent drift that some new intents might emerge and some old intent might become unpopular. Besides, diversity and redundancy issues are not carefully studied in sub-intent ranking with respect to the coverage of the intents.

3 Task formulation

3.1 NTCIR-11 IMine task

In the NTCIR-11 IMine Task (see footnote 2), participants are expected to generate a two-level hierarchy of underlying intents by analysis into the provided document collection, user behavior data set or other kinds of external data sources. Besides the hierarchy of intents, a ranking list of all first-level intents and a separate ranking list of all second-level intents should also be returned for each query.

A list of query suggestions/completions collected from popular commercial search engines are provided as possible intent candidates, including query suggestions collected from Bing, Google, Sogou, Yahoo! and Baidu,[6] query dimensions generated by (Dou et al. 2011b) from search engine result page,[7] related Web search queries generated by (Liu et al. 2011) from Sogou log data.[8]

3.2 Task reformulation

In this section, we reformulate the NTCIR-11 IMine task with a more general formalization.

A candidate *Text Fragment* can be any text fragment related to the query. *Candidate sub-intent*s represent the text fragments from which the intents of a query are mined. An example is shown in the upper part of Fig. 1. *Intent Hierarchy* represents the tree structure of a given query's intents. An two-level intent hierarchy example is shown in the lower part of Fig. 1. The first-layer of intent hierarchy consists of *Intents* and the second-layer consists of *Sub-intents.*

Based on above terminologies, the two main tasks of our work can be formulated as follows.

Task 1 (Query intent mining (QIM)) *Given a query q associated with a set of Candidate Sub-intents (text fragments related to q)* $\mathcal{C} = \{SI_1, SI_2, \ldots, SI_n\}$, *QIM task attempts to generate a two-level intent hierarchy, expressed as* $\mathcal{I}_q = \{I_1, I_2, \ldots\}$. *Each node of the first layer is referred to as an Intent* $I_l = \{\ldots, SI_i, SI_j, \ldots\}$, *while each node of the second layer is a Sub-intent* SI_i.

After *QIM*, each query can be interpreted as a two-level intent hierarchy. An instance of a two-level intent hierarchy is shown in Fig. 1, which is composed of two levels: the intent level and the sub-intent level. The intents represent specific objects or events of the query, where four intents (objects) of the Chinese query "先知 (Prophet)" are listed: "宗教先知 (Prophets in religion)", "先知 (The Chinese name of an actor in Dota game)", "先知电子狗 (Prophet electronic dog)" and "先知电影 (Movies about prophets)". Each intent can be described as a set of sub-intents. The sub-intents indicate the properties of the objects or events. For example, users may be interested in specific "prophets in religion" like "Jesus Christ" or "Abraham", or they may look for some properties of the movie "Knowing" like "actor Nicolas Cage" and "director Alex Proyas".

[6] http://www.thuir.cn/imine/IMine.QuerySuggestion.7z.

[7] http://www.thuir.cn/imine/IMine.QueryDimensions.7z.

[8] http://www.thuir.cn/imine/IMine.RelatedQueries.7z.

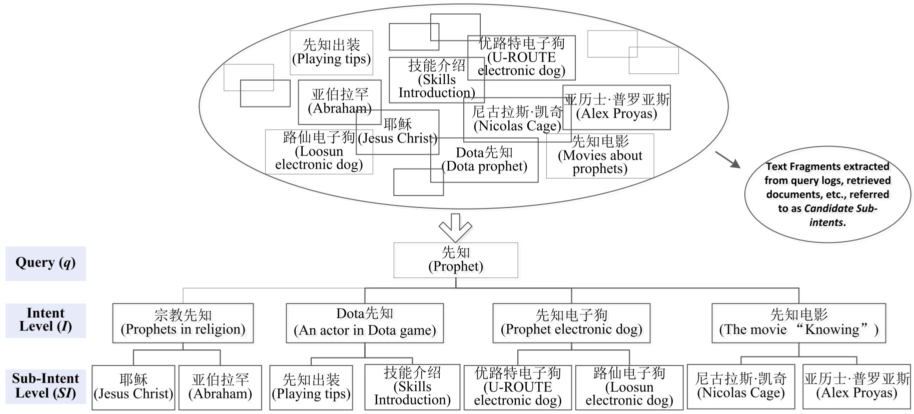

先知出装 (Playing tips)
技能介绍 (Skills Introduction)
优路特电子狗 (U-ROUTE electronic dog)
亚伯拉罕 (Abraham)
耶稣 (Jesus Christ)
尼古拉斯·凯奇 (Nicolas Cage)
亚历士·普罗亚斯 (Alex Proyas)
路仙电子狗 (Loosun electronic dog)
Dota先知 (Dota prophet)
先知电影 (Movies about prophets)
Text Fragments extracted from query logs, retrieved documents, etc., referred to as *Candidate Sub-intents.*
Query (*q*)
先知 (Prophet)
Intent Level (*I*)
宗教先知 (Prophets in religion)
Dota先知 (An actor in Dota game)
先知电子狗 (Prophet electronic dog)
先知电影 (The movie "Knowing")
Sub-Intent Level (*SI*)
耶稣 (Jesus Christ)
亚伯拉罕 (Abraham)
先知出装 (Playing tips)
技能介绍 (Skills Introduction)
优路特电子狗 (U-ROUTE electronic dog)
路仙电子狗 (Loosun electronic dog)
尼古拉斯·凯奇 (Nicolas Cage)
亚历士·普罗亚斯 (Alex Proyas)

Fig. 1 An example of the two-level intent hierarchy for the query "Prophet"

In our study, it should be noted that: (1) Not all candidate sub-intents are assigned to intents (or become sub-intents), i.e., $\mathcal{S} = \bigcup_{l=1}^{k} I_l \subseteq \mathcal{C}$. Actually, only candidate sub-intents ranked in the top list of each intent set become sub-intents. (2) Each sub-intent cannot belong to any two different intents, i.e., $I_l \cap I_m = \emptyset$ for $I_l, I_m \in \mathcal{I}_q$ and $l \neq m$.

Similar to other clustering tasks, for *QIM*, the central problem is to design an effective similarity function to measure the similarity between each pair of candidate sub-intents.

Task 2 (Query intent ranking (QIR)) *Given a query q and its two-level intent hierarchy* $\mathcal{I}_q = \{I_1, I_2, \ldots, I_k\}$. *The QIR task attempts to give a proper rank of (1) all of the intents in* $\mathcal{I}_q$ *and (2) all of the sub-intents* $S = \bigcup_{l=1}^{k} I_l$.

According to our definition, QIR task involves two ranking sub-tasks: rankings at the intent level and at the sub-intent level. Furthermore, ranking at the intent level should be handled firstly, as the ranking of the sub-intents depends on their intents: (1) A sub-intent could be important if its intent is important; (2) The sub-intents ranked in top positions should be diversified over intents.

4 Query intent mining

As discussed previously, designing an effective similarity function is the central problem in the *QIM* task. In this session, we measure the similarity between each pair of candidate sub-intents with consideration of three feature classes: Click Through-based Similarity, Search Result-based Similarity, and Query-based Similarity.

4.1 Click through-based similarity

Many text fragments are queries issued by users, and thus their similarity can be reflected from their users' click-through behaviors. Specially, the queries are likely to share a same intent if same URLs associated with these queries are clicked. In this study, we refer to such phenomenon as "*Same-Intent-Co-Click* (*SICC*)" as shown in Fig. 2. The *SICC* phenomenon can be interpreted as: most documents usually focus on a main topic. Thus

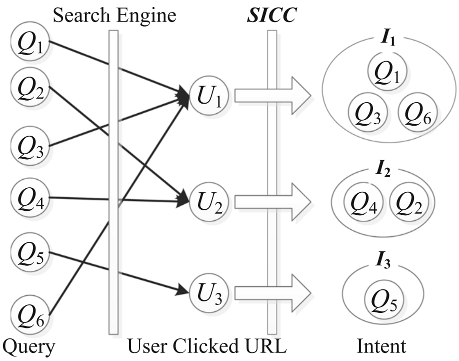


Fig. 2 The *SICC* phenomenon. *Note* *Q*, *U* and *I* represent query, URL and intent respectively. The *U* represents the user clicked URL returned by search engines

different queries submitted by multiple users reflect the same intent if most users click on the same URL to look for a same topic.

We experimentally validate the assumption of the *SICC* phenomenon based on the SogouQ dataset.[9] SogouQ is a dataset of 1-month Chinese query logs collected from the Chinese commercial search engine Sogou[10] in June 2012. It is officially offered by NTCIR INTENT-1, INTENT-2 and IMine task as a part of the Chinese dataset. Because it contains user queries that are not a part of the NTCIR-11 IMine task, we firstly filtered out 4270 queries (each query is a superset of the corresponding query of IMine) and 17232 URLs. Then, we manually labeled intents of these queries.[11] Finally, we analyzed the number of intents and the number of different queries for per URL, as shown in Fig. 3.

Figure 3a shows the distribution of the number of intents (vertical axis) with respect to the number of different queries (horizontal axis) that users click the same URL. We can see that there is a quite low positive correlation between them: the number of intents for one URL does not increase obviously with the growth of the number of different queries. That is, though different users use different queries when clicking a same URL, but the average number of intents for the clicked URL is less than 3. Figure 3b shows the percentage of URLs (vertical axis) with respect to the number of different queries (horizontal axis) per URL. We can see that, for more than 92.7 % URLs, users issue less than seven different queries when clicking the same URL. Combining Fig. 3a, b, we can conclude that, for more than 92.7 % URLs, the average number of intents behind user queries is less than 1.5, which is a strong statistical support for *SICC* phenomenon.

Our first similarity function is based on *SICC* phenomenon. Let SI_i and SI_j be any pair of candidate sub-intents. Let $\mathbf{U}_{SI_i}^{\Delta t}$ and $\mathbf{U}_{SI_j}^{\Delta t}$ be the click-through vectors of SI_i and SI_j during time Δt respectively. Each of them is composed of a vector of URLs that users clicked with the text fragments as queries. Thus the similarity between SI_i and SI_j can be measured as the similarity between $\mathbf{U}_{SI_i}^{\Delta t}$ and $\mathbf{U}_{SI_j}^{\Delta t}$. A frequently used metric is *Cosine*:

$$\Psi_1(SI_i, SI_j; \Delta t) = \frac{\mathbf{U}_{SI_i}^{\Delta t\top} \mathbf{U}_{SI_j}^{\Delta t}}{||\mathbf{U}_{SI_i}^{\Delta t}|| \times ||\mathbf{U}_{SI_j}^{\Delta t}||} \tag{1}$$

[9] http://www.sogou.com/labs/dl/q-e.html.

[10] http://www.sogou.com/.

[11] Because not all queries in Sogou logs are labeled in the official annotation results, we manually annotated the remaining queries according to the official annotation.

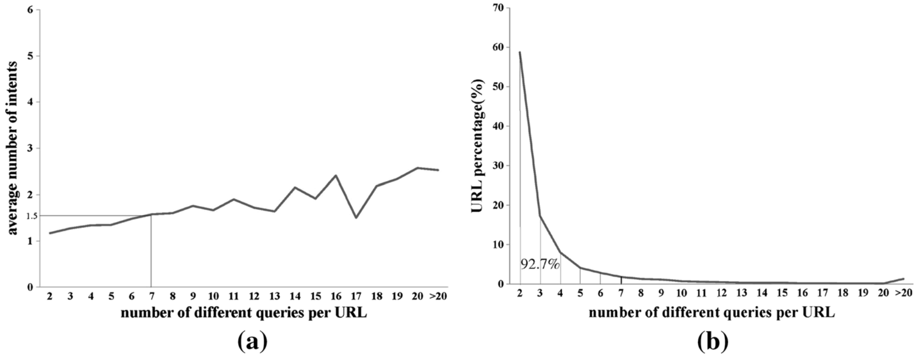


Fig. 3 *Same-Intent-Co-Click* (*SICC*) phenomenon analyzed with SogouQ (see footnote 9). **a** Average number of intents of the queries per URL. **b** Average number of different queries per URL

where "$\top$" is the transpose operator for a given vector or matrix, and "$||\cdot||$" is the 2-norm of a given vector. According to Eq. (1), the higher the $\Psi_1(SI_i, SI_j; \Delta t)$ is, the more likely SI_i and SI_j belong to a same intent.

We also adopt the approach proposed by Craswell and Szummer as another similarity $\Psi_2(SI_i, SI_j; \Delta t)$ (Craswell and Szummer 2007). $\Psi_2(SI_i, SI_j; \Delta t)$ is computed as follows. First, we build the Query-URL bipartite graph and transition probability matrix. Then, we extend it by adding self-transitions to the nodes. Finally, we perform the random walk and use the convergent transition probability between SI_i and SI_j as the similarity.

4.2 Search result-based similarity

The next three similarity functions are based on another common phenomenon that different queries associated with similar search results may imply the same intent. In this study, we refer to such phenomenon as "*Same-Intent-Similar-Rank* (*SISR*)" as shown in Fig. 4.

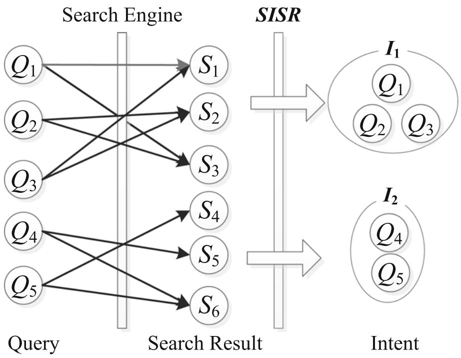
We also conducted experimental analysis on SogouQ (see footnote 9) for this phenomenon. First, we labeled intents of the corresponding queries in the Sogou logs and submitted them to the Google Search Engine.[12] Then for each pair of queries, we count the number of common documents among their top-N search results, where N varies from 5 to 100, as shown in Fig. 5. From the figure we can see that queries belonging to the same intent tend to have much more common documents in the search results than those belonging to different intents.

Given a query (a candidate sub-intent), its search results generally involve two aspects: (1) the contents of retrieved documents, and (2) the ranking of retrieved documents. Correspondingly, we define three search result-based similarity functions, of which the first two similarity functions are defined based on the content of retrieved documents, and the last one is based on the ranking of the documents.

Similarity based on the contents of retrieved documents For a pair of queries (candidate sub-intents) SI_i and SI_j, let $D_i^{\Delta t}$ and $D_j^{\Delta t}$ be the sets of their retrieved documents with time constraint Δt. In our experiments, we just used the snapshots of the top-N documents in the Google Search Engine (see footnote 12) instead of the full list of the documents for simplification.

[12] https://www.google.com.

Fig. 4 The *SISR* phenomenon. *Note Q*, *S*, and *I* represent query, search result and intent respectively. The *R* represents the search results returned by search engines (regardless of whether users click it or not)



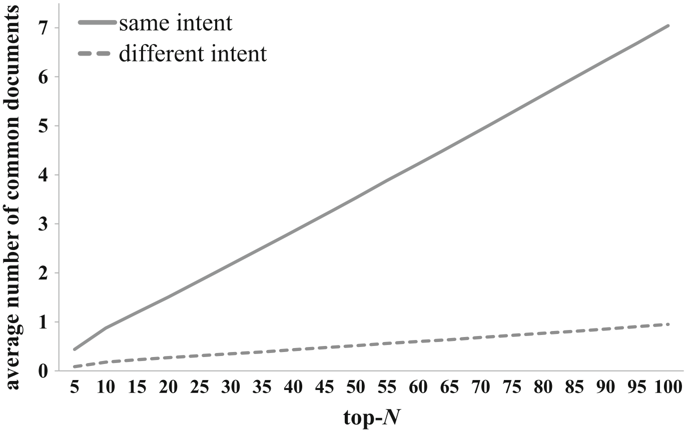


Fig. 5 *Same-Intent-Similar-Rank* (*SISR*) phenomenon analysis. The *horizontal axis* top-*N* denotes the top *N* search results, the *vertical axis* denotes the average number of common documents between all query-pairs' search results

Vector Space Model (VSM) and Language Model (LM) are two well-known representations of the documents. VSM regards each document as a bag of words and thus the document can be represented as a vector of word occurrences. Then, the similarity between two documents is evaluated with the Cosine metric. VSM performs well on tasks that involve measuring the similarity between words, phrases, and documents (Pantel and Lin 2002; Rapp 2003; Turney et al. 2003; Manning et al. 2008). LM recognizes each document as a sequence of words and assigns a probability to each permutation of the words. LM is also frequently used to evaluating the similarity between texts (Metzler et al. 2007; Erkan 2006; Kurland 2006). But different from VSM, LM evaluates the similarity of two documents from the perspective of probability distributions, i.e., the similarity or distance of two documents' language models.

From the perspective of VSM, the sets of the retrieved documents $D_i^{\Delta t}$ and $D_j^{\Delta t}$ during time Δt of SI_i and SI_j can be represented as vectors of words $\mathbf{W}_{D_i}^{\Delta t}$ and $\mathbf{W}_{D_j}^{\Delta t}$ respectively, and the similarity between SI_i and SI_j can be measured as the similarity between $\mathbf{W}_{D_i}^{\Delta t}$ and $\mathbf{W}_{D_j}^{\Delta t}$, which can be calculated with the cosine metric:

$$\Psi_3(SI_i, SI_j; \Delta t) = \frac{\mathbf{W}_{D_i}^{\Delta t \top} \mathbf{W}_{D_j}^{\Delta t}}{||\mathbf{W}_{D_i}^{\Delta t}|| \times ||\mathbf{W}_{D_j}^{\Delta t}||} \tag{2}$$

From the perspective of LM, through considering the common documents $D_i^{\Delta t} \cap D_j^{\Delta t}$, the similarity between SI_i and SI_j can be measured by the cross entropy between their probability distributions $P(d \mid SI_i)$ and $P(d \mid SI_j)$.

$$\Psi_4(SI_i, SI_j; \Delta t) = -\sum_{d \in D_i^{\Delta t} \cap D_j^{\Delta t}} P(d \mid SI_i) \cdot \ln P(d \mid SI_j) \tag{3}$$

where $P(d \mid SI)$ can be calculated with the language model (Manning et al. 2008).

Similarity based on the ranking of retrieved documents Given a pair of candidate sub-intents (queries) SI_i and SI_j, still let $D_i^{\Delta t}$ and $D_j^{\Delta t}$ be the sets of their retrieved documents, and let $\pi_i^{\Delta t}$ and $\pi_j^{\Delta t}$ be the rankings on their common document set $D_i^{\Delta t} \cap D_j^{\Delta t}$. From the angle of the document ranking, the similarity between SI_i and SI_j can be measured as the similarity between $\pi_i^{\Delta t}$ and $\pi_j^{\Delta t}$, which is generally calculated based on the Kendall's τ rank correlation coefficient (Kendall 1938; Marden 1996) on $D_i^{\Delta t} \cap D_j^{\Delta t}$.

$$\Psi_5(SI_i, SI_j; \Delta t) = \frac{1}{2} + \frac{1}{2}\tau(\pi_i^{\Delta t}, \pi_j^{\Delta t}) = \frac{1}{2} + \frac{N_c - N_d}{N(N-1)} \tag{4}$$

where $\tau(\pi_i^{\Delta t}, \pi_j^{\Delta t})$ is the value of the Kendall's τ correlation coefficient between two rankings $\pi_i^{\Delta t}$ and $\pi_j^{\Delta t}$, N_c and N_d are the numbers of their concordant pairs and discordant pairs respectively, and $N = |D_i^{\Delta t} \cap D_j^{\Delta t}| = N_c + N_d$ is the cardinality of the common document set $D_i^{\Delta t} \cap D_j^{\Delta t}$. Obviously, $\Psi_5(SI_i, SI_j; \Delta t)$ varies from 0 to 1. In particular, $\Psi_5(SI_i, SI_j; \Delta t) = 0$ if $\pi_i^{\Delta t} \cap \pi_j^{\Delta t} = \emptyset$ or $\pi_i^{\Delta t}$ is a reverse ranking of $\pi_j^{\Delta t}$ and thus $N_c = 0$ and $N_d = \frac{1}{2}N(N-1)$, while $\Psi_5(SI_i, SI_j; \Delta t) = 1$ if $\pi_i^{\Delta t}$ is the same as $\pi_j^{\Delta t}$ and thus $N_c = \frac{1}{2}N(N-1)$ and $N_d = 0$.

4.3 Query-based similarity

We further explore more similarity functions which directly measure the similarity of two text fragments (candidate sub-intents).

Syntactic similarity Syntactic similarity describes the string match between two sub-intents. Ψ_6 is such metric that takes exact term match and term sequence into account.

$$\Psi_6(SI_i, SI_j) = \frac{SC(SI_i, SI_j) + SC(SI_j, SI_i)}{2} \tag{5}$$

where

$$SC(SI_i, SI_j) = \frac{1}{|SI_i|} \sum_{p=1}^{|SI_i|} \frac{|SI_j| - \min\{|SI_j|, |p - p'| : p' \in Pos(SI_j, SI_i^p)\}}{|SI_j|}$$

where SI_i^p represents the term at position p of sub-intent SI_i. If SI_j contains the term SI_i^p, $Pos(SI_j, SI_i^p)$ is the set of all positions of SI_i^p in SI_j, otherwise $Pos(SI_j, SI_i^p)$ is null. If $Pos(SI_j, SI_i^p)$ is null, then $|p - p'|$ is null too.

Ψ_7 is another syntactic similarity that computes the cosine similarity of the term vectors of the sub-intents. Different from Ψ_6, it ignores term sequence.

$$\Psi_7(SI_i, SI_j) = \frac{\mathbf{W}_{SI_i} \cdot \mathbf{W}_{SI_j}}{||\mathbf{W}_{SI_i}|| \times ||\mathbf{W}_{SI_j}||} \tag{6}$$

where $\mathbf{W}_{SI}$ is the term vector of the sub-intent SI.

Semantic similarity While the sub-intents may not have direct term overlap, they may be similar semantically. To address this, we also define some semantic similarities. Different from measuring the semantic similarity of two documents, Language Model-like metrics are not suitable for the short text characteristics. The first similarity function we defined is based on the experience that words of the same intent are more frequently co-occur than that of different intents.

$$\Psi_8(SI_i, SI_j) = \frac{\sum_{w_1 \in SI_i; w_2 \in SI_j; w_1 \neq w_2} cooccur(w_1, w_2)}{\sum_{w_1, w_2 \in SI_i \cup SI_j \cup q; w_1 \neq w_2} cooccur(w_1, w_2)} \tag{7}$$

where $cooccur(w_1, w_2)$ is the frequency when w_1 and w_2 co-occur in the same sentence of search result snapshots or same queries in the search logs.

The next two similarity functions are based on HowNet.[13] HowNet is a Chinese lexical database similar to WordNet.[14] A variety of semantic similarities are implemented based on information found in the lexical database. The following two similarities are adopted in this paper, of which $SIM^{Liu}(w_m, w_n)$ and $SIM^{Xia}(w_m, w_n)$ are based on algorithms proposed by Liu (Liu and Li 2002) and Xia (Xia 2007) respectively.

$$\begin{aligned} \Psi_9(SI_i, SI_j) &= \max\{SIM^{Liu}(w_m, w_n) | w_m \in SI_i \backslash q, w_n \in SI_j \backslash q, w_m \neq w_n\} \\ \Psi_{10}(SI_i, SI_j) &= \max\{SIM^{Xia}(w_m, w_n) | w_m \in SI_i \backslash q, w_n \in SI_j \backslash q, w_m \neq w_n\} \end{aligned} \tag{8}$$

4.4 Learning-based similarity integration

Among above ten similarity functions, the first click-through-based and three search result-based functions are time-dependent, while the other functions are time-independent. Δt may refer to different time granularity such as 'day', 'week', 'month', 'year' or time intervals automatically segmented according to events period detection (Qian et al. 2013). All of the similarity functions have a same scope of values from 0 to 1. With the proposed ten similarity functions, the final similarity function can be straightforwardly obtained in a linear integration. Generally, most existing work utilized heuristic methods to design the coefficients of the functions, because they usually use less than three functions for integration, which is relatively easy to decide the best values heuristically (Wen et al. 2001; Xue et al. 2011; Hu et al. 2012; Qian et al. 2013). In our study, we consider ten similarity functions, which is much more difficult to optimize coefficients heuristically.

We propose a learning-based integration method. Specially, given two candidate sub-intents SI_i and SI_j, let $\Psi(SI_i, SI_j; \Delta t) = [\psi_1(SI_i, SI_j; \Delta t), \ldots, \psi_{10}(SI_i, SI_j; \Delta t)]^\top$ be the vector of the similarity metrics, where the elements are the ten similarity metrics respectively. Thus the linearly integrated similarity metric can be defined as follows:

$$SIM(SI_i, SI_j; \Delta t) = \boldsymbol{\theta}^\top \Psi(SI_i, SI_j; \Delta t) \tag{9}$$

[13] http://www.keenage.com.

[14] http://wordnet.princeton.edu.

where $\boldsymbol{\theta} = [\vartheta_1, \ldots, \vartheta_{10}]^\top$ is the coefficient vector, $0 \leq \vartheta_1, \ldots, \vartheta_{10} \leq 1$ and $\sum_{m=1}^{10} \vartheta_m = 1$ hold. Since $0 \leq \psi_m(SI_i, SI_j; \Delta t) \leq 1$ for $m = 1, \ldots, 10$, the integrated similarity $0 \leq SIM(SI_i, SI_j; \Delta t) \leq 1$.

Let Q be the training dataset, which is comprised of a set of queries. Each query $q \in Q$ has a set of intents $\mathcal{I}_q$, and each intent I_l is comprised of a set of sub-intents. In our learning-based integration method, given a query q, we try to maximize the similarity between the sub-intents in same intents while minimize the similarity between the sub-intents in different intents. Thus the loss function can be defined as follows:

$$\begin{aligned} \arg\min\nolimits_{\boldsymbol{\theta}} \quad & F(\boldsymbol{\theta}) \\ s.t. \quad & \sum_{m=1}^{10} \vartheta_m = 1 \\ & \vartheta_m \geq 0, \quad m = 1, 2, \ldots, 10 \end{aligned} \tag{10}$$

where

$$\begin{aligned} F(\boldsymbol{\theta}) &= -\log\left(\prod_{q\in Q}\prod_{I_l\in\mathcal{I}_q}\frac{\sum_{SI_i,SI_j\in I_l,i\neq j} SIM(SI_i,SI_j;\Delta t)}{\sum_{\overline{I}_l\in\mathcal{I}_q}\sum_{SI_i\in I_l,SI_j\in\overline{I}_l,i\neq j} SIM(SI_i,SI_j;\Delta t)}\right) + \frac{\eta}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} \\ &= -\sum_{q\in Q}\sum_{I_l\in\mathcal{I}_q}\log\left(\frac{\sum_{SI_i,SI_j\in I_l,i\neq j}\boldsymbol{\theta}^\top\Psi(SI_i,SI_j;\Delta t)}{\sum_{\overline{I}_l\in\mathcal{I}_q}\sum_{SI_i\in I_l,SI_j\in\overline{I}_l,i\neq j}\boldsymbol{\theta}^\top\Psi(SI_i,SI_j;\Delta t)}\right) + \frac{\eta}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} \\ &= \sum_{q\in Q}\sum_{I_l\in\mathcal{I}_q}\left[\log\left(\sum_{\overline{I}_l\in\mathcal{I}_q}\sum_{SI_i\in I_l,SI_j\in\overline{I}_l,i\neq j}\boldsymbol{\theta}^\top\Psi(SI_i,SI_j;\Delta t)\right)\right] \\ &\quad -\sum_{q\in Q}\sum_{I_l\in\mathcal{I}_q}\left[\log\left(\sum_{SI_i,SI_j\in I_l,i\neq j}\boldsymbol{\theta}^\top\Psi(SI_i,SI_j;\Delta t)\right)\right] + \frac{\eta}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} \end{aligned} \tag{11}$$

We use the Lagrange multiplier method to solve the constrained optimization problem. Let

$$L(\boldsymbol{\theta},\lambda) = F(\boldsymbol{\theta}) + \lambda(1 - \sum_{m=1}^{10}\vartheta_m)$$

Then we can straightforwardly solve it with the gradient descent method (Calamai and Moré 1987). The gradient of the $L(\boldsymbol{\theta},\lambda)$ with respect to $\boldsymbol{\theta}$ and λ is as follows:

$$\left\{\begin{array}{ll} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta},\lambda) = & \sum\limits_{q\in Q}\sum\limits_{I_l\in\mathcal{I}_q}\left(\dfrac{\sum_{\overline{I}_l\in\mathcal{I}_q}\sum_{SI_i\in I_l,SI_j\in\overline{I}_l,i\neq j}\Psi(SI_i,SI_j;\Delta t)}{\sum_{\overline{I}_l\in\mathcal{I}_q}\sum_{SI_i\in I_l,SI_j\in\overline{I}_l,i\neq j}\boldsymbol{\theta}^\top\Psi(SI_i,SI_j;\Delta t)}\right) \\ & -\sum\limits_{q\in Q}\sum\limits_{I_l\in\mathcal{I}_q}\left(\dfrac{\sum_{SI_i,SI_j\in I_l,i\neq j}\Psi(SI_i,SI_j;\Delta t)}{\sum_{SI_i,SI_j\in I_l,i\neq j}\boldsymbol{\theta}^\top\Psi(SI_i,SI_j;\Delta t)}\right) + \eta\boldsymbol{\theta} - \lambda\mathbf{1}_{10} \\ \dfrac{\partial}{\partial\lambda}L(\boldsymbol{\theta},\lambda) = & 1 - \sum\limits_{m=1}^{10}\vartheta_m \end{array}\right.$$

where $\mathbf{1}_{10}$ is a 10-dimensional vector of which all of the elements are 1. Thus the update formula of $\boldsymbol{\theta}$ and λ are as follows:

$$\begin{aligned}\boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} - \alpha \bigtriangledown_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \lambda) \\ \lambda &\leftarrow \lambda - \alpha \frac{\partial}{\partial \lambda} L(\boldsymbol{\theta}, \lambda)\end{aligned} \tag{12}$$

Note that we have another constraint $\vartheta_m \geq 0$ for $m = 1, 2, \ldots, 10$. Thus, in each iteration, just simply set $\vartheta_m = 0$ if any $\vartheta_m < 0$ happens for $m = 1, \ldots, 10$ after updating the values of $\boldsymbol{\theta}$.

4.5 Clustering algorithm

```
ALGORITHM 1: Sub-intent Clustering Algorithm
Input: A given query $q$ with a set of candidate sub-intents such that $\mathcal{C} = \{SI_1, SI_2, ...SI_n\}$, related user search logs $\mathcal{L}$ and
       HowNet $\mathcal{H}$.
Output: A hierarchical intent model $\mathcal{I}_q = \{I_1, I_2, \cdots, I_k\}$, where each intent $I_l = \{SI_{l_1}, SI_{l_2}, \cdots, SI_{l_b}\}$

Step 1: Calculate $SIM(SI_i, SI_j; \Delta t)$ between each pair of sub-intents of $q$ and build the similarity matrix $\mathbf{M}$;
Step 2: Run $AP$ clustering algorithm using the similarity matrix $\mathbf{M}$ as the input;
Step 3: Merge two clusters of the result of $AP$ clustering algorithm if the group average similarity of the two clusters is
larger than threshold $\epsilon$;
Step 4: Finish when no clusters can be merged;
Step 5: Regard each cluster as an intent $I_l = \{SI_{l_1}, SI_{l_2}, \cdots, SI_{l_b}\}$ and the exemplar of each cluster as cluster label;
Step 6: Output all clusters such that $\mathcal{I}_q = \{I_1, I_2, \cdots, I_k\}$.
```

With the integrated similarity function, we generate a two-level intent hierarchy for a given query with an affinity propagation(Frey and Dueck 2007)-based clustering method. We choose this clustering approach because affinity propagation does not need to predefine the number of clusters and it recently has received tremendous attentions from different areas, including image (Wang et al. 2013), text (Sun and Guo 2014), stream (Zhang et al. 2014), and hierarchical (Kazantseva and Szpakowicz 2014) clustering tasks for its effectiveness. The intent mining algorithm is shown in Algorithm 1, where the parameter $\epsilon = 0.5$ (which is the best performance setting in our *QIM* experiments). Each cluster represents one intent of the query.

5 Query intent ranking

Given a query associated with a set of intents and sub-intents, it is crucial to order them properly, so that search engines are able to adjust the search results or make query suggestion/expansion for fulfillment of users' information needs. In this section, we present ranking models for the intent and sub-intent levels respectively.

5.1 Intent ranking

In intent ranking, the main challenge is the intent drift problem, that the importance of the intents is time-sensitive and evolves over time. Some new intents emerge and receive great attentions while some old ones become unpopular. However, existing studies mostly focus on understanding user intents from a static viewpoint.

In this study, we construct a time-sensitive importance function $s(I_l; q, \Delta t){:}\mathcal{I}_q \rightarrow \mathbb{R}$ to calculate the ranking scores of the intents for the given query q at Δt.

Intent ranking objective function The importance of the intents involves two issues: cluster quality and intent relevance, as follows:

$$s(I_l; q, \Delta t) = q(I_l; \Delta t) \cdot r_T(q, I_l; \Delta t) \tag{13}$$

where $q(I_l; \Delta t)$ measures the clustering quality of intent I_l, $r_T(q, I_l; \Delta t)$ is the temporal relevance metric of intent I_l to the query q at Δt.

Clustering quality of intents Since each intent is actually a cluster of sub-intents, the quality of an intent can be measured with the metric of the clustering algorithms, i.e., the more close the sub-intents in the intent are, the higher the intent quality is. Specially, given an intent $I_l = \{SI_{l_1}, SI_{l_2}, \ldots, SI_{l_b}\}$ composed of l_b sub-intents, the quality metric of an intent is defined as follows:

$$q(I_l; \Delta t) = Z_Q \cdot \frac{\sum_{SI_{l_m}, SI_{l_n} \in I_l, m \neq n} SIM(SI_{l_m}, SI_{l_n}; \Delta t)}{l_b(l_b - 1)} \tag{14}$$

where Z_Q is a normalization factor which normalizes the value of the intent quality into the scale from 0 to 1.

Temporal relevance of intents We define the temporal relevance of the intents from the perspectives of the TF-IDF model and the language model respectively.

TF-IDF model (Manning et al. 2008) is a standard weighting scheme for weighting the relevance of the terms in information retrieval, which is calculated as product of the term frequency and the inverse document frequency. The term frequency *tf*(*w*, *d*) of word *w* in document *d* is defined as (the logarithm of) the number of times that *w* occurs in *d*. It positively contributes to the relevance of *d* to *w*. The inverse document frequency *idf* (*w*, *D*) of word *w* in corpus *D* measures the rarity of *w* in *D*, which is defined as (the logarithm of) the inverse of the document frequency *df*(*w*, *D*), i.e., the number of documents that *w* occurs in the corpus *D*. If *w* is rare, then the documents containing *w* are more relevant to *w*.

Given a query q and an intent I_l, let W_l be the set of words in I_l, and $D(q, \Delta t)$ be the set of the documents retrieved by q with time constraint Δt. In our experiments, we just used the snapshots of the top-*N* documents in the Google Search Engine (see footnote 12). Thus from the perspective of the TF-IDF model, the temporal relevance of the intents $r_T(q, I_l; \Delta t)$ can be calculated as the average TF-IDF weighting scores of the words in I_l and the corpus $D(q, \Delta t)$. Formally,

$$r_T(q, I_l; \Delta t) = Z_T \cdot \frac{\sum_{w \in W_l} tf(w, D(q, \Delta t)) \cdot idf(w, D(q, \Delta t))}{|W_l|} \tag{15}$$

where Z_T is a normalization factor which normalizes the value of $r_T(q, I_l; \Delta t)$ into the scale from 0 to 1, $tf(w, D(q, \Delta t))$ and $idf(w, D(q, \Delta t))$ utilize their logarithm formulations (Manning et al. 2008).

5.2 Sub-intent ranking

Sub-intents represent users' detailed information needs, such as the properties of the objects and events. Thus discovery of top-*k* important sub-intents is a central problem for many applications in information retrieval, e.g. query suggestion and search diversification.

Sub-intent ranking objective function Given a query q, all of its intents $\mathcal{I}_q$ and sub-intents $\mathcal{S} = \bigcup_{I_l \in \mathcal{I}_q} I_l$, the sub-intent ranking task aims to generate a top-*k* ranking list $\pi \subseteq \mathcal{S}$ according to their importance at Δt with consideration of temporal issues and diversity issues. The objective function is defined as follows:

$$\arg\max_{\pi \subseteq \mathcal{S}, |\pi| = k} L(\pi; \mathcal{I}_q, q, \Delta t) = \arg\max_{\pi \subseteq \mathcal{S}, |\pi| = k} \sum_{I_l \in \mathcal{I}_q} s(I_l; q, \Delta t) \cdot \Phi(\pi; I_l, q, \Delta t) \tag{16}$$

where $s(I_l; q, \Delta t)$ is the importance of intent I_l defined in Formula 13; $\Phi(\pi; I_l, q, \Delta t)$ evaluates the extent to which the ranking list π covers intent I_l. Thus, *the explanation for our objective (Formula16) is, we try to maximize the coverage of intents in terms of their importance in top-k sub-intents ranking list* π. $\Phi(\pi; I_l, q, \Delta t)$ is defined as follows:

$$\Phi(\pi; I_l, q, \Delta t) = 1 - \prod_{SI_i \in \pi} (1 - P(SI_i; \pi, I_l, q, \Delta t)) \tag{17}$$

$P(SI_i; \pi, I_l, q, \Delta t)$ is the probability that sub-intent SI_i covers intent I_l at Δt.

$$P(SI_i; \pi, I_l, q, \Delta t) = imp(SI_i; I_l, q, \Delta t) \cdot nov(SI_i; \pi, \Delta t) \tag{18}$$

Thus $1 - P(SI_i; \pi, I_l, q, \Delta t)$ is the probability that SI_i fails to cover I_l. According to the independence assumption of Naïve Bayes, $\prod_{SI_i \in \pi}(1 - P(SI_i; \pi, I_l, q, \Delta t))$ represents the probability that all sub-intents in π fail to cover I_l. Then, $1 - \prod_{SI_i \in \pi}(1 - P(SI_i; \pi, I_l, q, \Delta t))$ is the probability that at least one sub-intent in π covers intent I_l.

Temporal relevance of sub-intents In Formula 18, $imp(SI_i; I_l, q, \Delta t)$ is the temporal relevance of SI_i at Δt, which involves two aspects: the temporal importance of SI_i for its intent I_l, and the relevance of SI_i for the query q. Formally,

$$imp(SI_i; I_l, q, \Delta t) = sim(SI_i, q; \Delta t) \cdot sim(SI_i, I_l; \Delta t) \tag{19}$$

where $sim(SI_i, q; \Delta t)$ is the temporal importance of SI_i for the query q, which can be evaluated using term frequency of SI_i in the set of retrieved documents $D(q, \Delta t)$ by q at Δt. Similar to previous settings, we used the snapshots of the top-N documents in the Google Search Engine (see footnote 12). Formally,

$$sim(SI_i, q; \Delta t) = \frac{\sum_{w \in SI_i} tf(w, D(q, \Delta t))}{|SI_i| \cdot |D(q, \Delta t)|} \tag{20}$$

$sim(SI_i, I_l; \Delta t)$ is the relevance of SI_i to I_l, which can be calculated as the similarity between SI_i and its intent I_l, i.e., the average similarity between SI_i and each sub-intent in I_l:

$$sim(SI_i, I_l; \Delta t) = \frac{1}{|I_l|} \sum_{SI_j \in I_l} SIM(SI_i, SI_j; \Delta t) \tag{21}$$

where $SIM(SI_i, SI_j; \Delta t)$ is calculated with Formula 9, and $|I_l|$ is the number of sub-intents in I_l.

Novelty of sub-intents $nov(SI_i; \pi, \Delta t)$ in Formula 18 is to measure the novelty of choosing SI_i into π, which is defined as the difference between SI_i and the other sub-intents in π. Formally,

$$nov(SI_i; \pi, \Delta t) = 1 - (1 - \zeta) \max_{SI_j \in \pi} SIM(SI_i, SI_j; \Delta t) \tag{22}$$

where ζ is a very small positive number to make sure that the minimal value of $nov(SI_i; \pi, \Delta t)$ is $\zeta \neq 0$ and thus $P(SI_i; \pi, I_l, q, \Delta t) \neq 0$. In our experiments, we set $\zeta = 1E^{-4}$.

The intuition of $nov(SI_i; \pi, \Delta t)$ is that π should cover as many intents as possible and simultaneously not contain similar sub-intents. For example, if π already includes the sub-intent “尼古拉斯· 凯奇先知电影 (‘Knowing’ starring Nicolas Cage)”, it should not contain the similar sub-intents like “

先知电影 (the movie ‘Knowing’)” any more.

Optimization As shown in Theorem 1, we prove that our objective function 16 is non-negative, monotone and sub-modular.

Theorem 1 *$L(\pi;\mathcal{I}_q,q,\Delta t)$ is a non-negative, monotone and sub-modular function.*

Proof It is obvious that $L(\pi;\mathcal{I}_q,q,\Delta t)$ is non-negative. Let π_1, π_2 be two arbitrary list of sub-intents related by $\pi_1 \subset \pi_2$. Let SI be a sub-intent satisfying $\pi_2 = \pi_1 \cup \{SI\}$.

$$\begin{aligned} & L(\pi_2;\mathcal{I}_q,q,\Delta t) - L(\pi_1;\mathcal{I}_q,q,\Delta t) \\ & \quad = \sum_{I_l \in \mathcal{I}_q} (\Lambda(I_l;q,\Delta t) \cdot (\Phi(\pi_2;I_l,q,\Delta t) - \Phi(\pi_1;I_l,q,\Delta t))) \\ & \quad = \sum_{I_l \in \mathcal{I}_q} \left(\Lambda(I_l;q,\Delta t) \cdot (V(SI) \cdot \prod_{SI_i \in \pi_1} (1 - V(SI_i))) \right) \end{aligned}$$

where $V(SI) = V(SI_i) = (imp(SI_i;I_l,q,\Delta t) \cdot nov(SI_i;\pi)) \in [0,1]$, $\Lambda(I_l;q,\Delta t) \geq 0$. So $L(\pi_2;\mathcal{I}_q,q,\Delta t) \geq L(\pi_1;\mathcal{I}_q,q,\Delta t)$ which means $L(\pi;\mathcal{I}_q,q,\Delta t)$ is monotone.

Let π_3 be any arbitrary list of sub-intents satisfying $\pi_1 \subset \pi_3$. Let SI' be a sub-intent not in π_3. Denote $\pi_1 \cup \{SI'\}$ with $\pi_1{}'$ and similarly $\pi_3 \cup \{SI'\}$ with $\pi_3{}'$.

$$\begin{aligned} f_1 & = L(\pi_1{}';\mathcal{I}_q,q,\Delta t) - L(\pi_1;\mathcal{I}_q,q,\Delta t) \\ & = \sum_{I_l \in \mathcal{I}_q} \left(\Lambda(I_l;q,\Delta t) \cdot (V(SI') \cdot \prod_{SI_i \in \pi_1} (1 - V(SI_i))) \right) \end{aligned}$$

Similarly, we can establish

$$\begin{aligned} f_2 & = L(\pi_3{}';\mathcal{I}_q,q,\Delta t) - L(\pi_3;\mathcal{I}_q,q,\Delta t) \\ & = \sum_{I_l \in \mathcal{I}_q} \left(\Lambda(I_l;q,\Delta t) \cdot (V(SI') \cdot \prod_{SI_i \in \pi_3} (1 - V(SI_i))) \right) \end{aligned}$$

Note that $V(SI_i) \in [0,1]$ and $\pi_1 \subset \pi_3$, we have $\prod_{SI_i \in \pi_1} (1 - V(SI_i)) \geq \prod_{SI_i \in \pi_3} (1 - V(SI_i))$. Therefore, we conclude that $f_1 \geq f_2$, i.e., the function $L(\pi;\mathcal{I}_q,q,\Delta t)$ is submodular. □

The value of Theorem 1 reflects in two aspects. First, according to the work done by Cornuejols et al. (1977) and Nemhauser et al. (1978), if an objective function is proved to be *non-negative*, *monotone* and *sub-modular* (which Theorem 1 does), then we can make a conclusion that the Greedy algorithm will achieve at least a $(1 - \frac{1}{e})$ approximation of the optimal solution. Because the parameter values (i.e., $\Delta t, \zeta$) are given before we run the Greedy algorithm, they will not influence above conclusion. The Greedy algorithm is shown in Algorithm 2. Initially the ranking π is empty, and iteratively select a sub-intent SI_i from $\mathcal{S} \backslash \pi$ that maximizes the Formula 16 at step i, and set its ranking position as i. Second, the practical running time of Greedy for this problem can be alleviated by *CELF* (Leskovec et al. 2007) and *CELF*++ (Goyal et al. 2011). For details of the two algorithms, we refer the readers to the papers (Leskovec et al. 2007; Goyal et al. 2011).

ALGORITHM 2: The Greedy Algorithm For Sub-intent Ranking

```
Input: A query $q$, its two-level intent hierarchy $\mathcal{I}_q$ (including sub-intents $\mathcal{S}$), and a time span $\Delta t$
Output: A top-$k$ ranking list $\pi$ of sub-intents

Step 1: Initially set the ranking $\pi = ()$ is empty at iteration $i = 0$;
Step 2: Repeat:;
Step 3: Set $r \leftarrow r + 1$ ;
Step 4: Select the sub-intent $SI_i \in \{\mathcal{S} \setminus \pi\}$ which maximizes the Formula 16, and set its ranking position as $r$:
$\pi \leftarrow \pi \succ SI_i$ ;
Step 5: Go to Step 2 until $r = k$;
Step 6: Output $\pi$.
```

6 Experiment

6.1 Experimental setup

Data sets To evaluate the performance of our method, we used the Chinese dataset of NTCIR-11 IMine (see footnote 2). *We choose this dataset because this is the only available dataset with real query logs, to the best of our knowledge*. The official dataset contains: (1) 50 Chinese queries. (2) Query logs SogouQ (see footnote 9). (3) The candidate sub-intents: Query suggestions collected from Sogou (see footnote 10), Google (see footnote 12), Yahoo!,[15] Bing[16] and Baidu;[17] Query dimensions generated by (Dou et al. 2011b) from search engine result pages; Related queries generated by (Liu et al. 2011) from SogouQ (see footnote 9).

For each candidate sub-intent, we collected Google search results belonging to the time span from January 2004 to July 2013. Specially, for each month, we issued each candidate sub-intent to Google Search with the condition of time range, and collected the top 500 results in the ranking list. For example, "Adobe" was submitted with the time condition "1 Jan, 2004–31 Jan, 2004" which can be specified in "Google Search Tools". The number of snapshots in the document dataset used in the experiment is about 8.5 million.

Evaluation measures Average accuracy is used to measure the quality of the hierarchical structure by whether the sub-intent is correctly assigned to the appropriate intent.

$$Accuracy = \frac{\sum_{l=1}^{|\mathcal{I}_q|} Correct(I_l)}{|\mathcal{I}_q|} \tag{23}$$

where $|\mathcal{I}_q|$ is the number of mined intents for a certain query. I_l denotes the lth intent. $Correct(I_l)$ is the percentage of correctly-assigned sub-intents for intent I_l. If intent I_l is not relevant to the query (annotated manually), then $Correct(I_l)$ should be 0.

D#-nDCG, a frequently-used measure for ranking diversification, is used to evaluate the quality of the intent/sub-intent ranking list by judging whether all important intents/sub-intents are found and ranked correctly (Sakai and Song 2011).

$$D\#\text{-}nDCG@k = \rho \cdot I\text{-}rec@k + (1 - \rho) \cdot D\text{-}nDCG@k \tag{24}$$

where ρ is set to 0.5 in this paper. *I-rec*@k is the intent recall at top k, i.e. percentage of intents found (for intent ranking) or covered by sub-intents (for sub-intent ranking). *D-nDCG*@k is computed by replacing the raw gain of *nDCG* with the global gain:

[15] https://www.yahoo.com/.

[16] https://www.bing.com/.

[17] http://www.baidu.com/.

$$GG(r) = \sum_{I_l} P(I_l|q) g_r(I_l) \tag{25}$$

where $g_r(I_l)$ is the gain value of intent/sub-intent at position r for intent I_l (ground truth).

H-measure[2] (Liu et al. 2014) is used to evaluate the performance of intent mining, intent ranking as well as sub-intent ranking.

$$H\text{-}measure = Accuracy \cdot (\tau \cdot D\#\text{-}nDCG@I + (1-\tau) \cdot D\#\text{-}nDCG@SI) \tag{26}$$

where *D#-nDCG@I* is the *D#-nDCG* of all intents in the ranking list, similarly, *D#-nDCG@SI* is the *D#-nDCG* of all sub-intents in the ranking list, and τ is set to 0.5.

Baseline Our first baseline is *TUTA*1 (Yu and Ren 2013), which is the best Chinese run in NTCIR-10 INTENT-2 task. We implement the approach according to their paper and apply it to the NTCIR-11 IMine dataset. In order to find out whether the proposed *SICC* and *SISR* features are actually useful, we use another baseline, namely *CONTENT-BASED*, which uses Cosine of the queries as the similarity and the proposed clustering approach in this paper. As for sub-intent ranking, we use IA-Select (Agrawal et al. 2009) and PM2 (Dang and Croft 2012) as baselines. We also choose the official results of IMine for comparison which includes 11 baselines (Liu et al. 2014). Note that our work is the first considering intent drift issue in intent and sub-intent ranking, to the best of our knowledge. However, we can not measure our study due to the lack of available data, baseline methods and evaluation measures. So we have to set Δt to full time (from Jan. 2004 to Dec. 2013 in this paper) in order to compare our methods with the baseline. Additionally, we make an interesting case study with intent drift in Sect. 6.3 to demonstrate the effectiveness of our study.

Parameter setting Fivefold cross validation is adopted to train the parameters of our approach with INTENT-1[18] and INTENT-2[19] data. Cross-validation is widely used in learning approaches to choose the values of parameters (Tibshirani and Tibshirani 2009). First, we randomly split the data into five pieces. Then, the parameters were estimated using fivefold cross-validation. Finally, the parameters were computed as the median of the five estimations (Chapelle et al. 2002). The final coefficient vector $\boldsymbol{\theta}$ for these ten similarity metrics with fivefold cross-validation is $\boldsymbol{\theta} = [0.12, 0.05, 0.11, 0.11, 0.14, 0.13, 0.12, 0.07, 0.08, 0.07]$. We also consider heuristical $\boldsymbol{\theta}$ which treats these 10 similarity metrics equally in the experiment.

6.2 Experiment results

The experiment results are shown in Tables 1, 2, 3 and 4. The first three tables compare our methods with *TUTA*1, and the last table compares our methods with the other baselines. *OurMethod*^{1} represents our method with heuristical $\boldsymbol{\theta}$; *OurMethod*^{2} represents our method with learned $\boldsymbol{\theta}$.

The intent mining result in Table 1 shows that both *OurMethod*^{1} and *OurMethod*^{2} outperform *TUTA*1, and *OurMethod*^{2} achieves the best performance. There are three reasons for the improvement: (1) Our methods consider three feature classes: click through-based similarity, search result-based similarity, and query-based similarity, while

[18] http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings9/NTCIR/Evaluations/INTENT/ntc9-INTENT-eval.htm.

[19] http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/NTCIR/Evaluations/INTENT/ntc10-INTENT2-eval.htm.

Table 1 Accuracy of intent mining

	Accuracy
*TUTA*1	0.417↓
CONTENT-BASED	0.148↓
OurMethod^1	0.526↓
OurMethod^2	**0.568**

↓ indicates *OurMethod*^{2} is statistically significantly (p value < 0.05 with two-tailed t test) better than this method. The *bold face* indicates the best performance

Table 2 *D#-nDCG* of intent ranking

	D#-nDCG				
	@1	@2	@3	@4	@5
*TUTA*1	0.483↓	0.453↓	0.341↓	0.419↓	0.396↓
OurMethod^1	0.605↓↓	0.601↓	0.551↓	0.503↓	0.576↓
OurMethod^2	**0.669**	**0.637**	**0.609**	**0.673**	**0.634**

↓ indicates *OurMethod*^{2} is statistically significantly (p value < 0.05 with two-tailed t test) better than this method. The *bold face* indicates the best performance

Table 3 *D#-nDCG* of sub-intent ranking

	D#-nDCG				
	@1	@5	@10	@20	@50
*TUTA*1	0.343↓	0.486↓	0.564	0.417	0.470↓
IA-Select	0.213↓	0.398↓	0.571	0.573	0.597
PM2	0.201↓	0.297↓	0.483	0.495↓	0.512↓
OurMethod^1	0.380↓	0.491↓	0.542↓	0.579↓	**0.638**
OurMethod^2	**0.411**	**0.545**	**0.576**	**0.609**	0.637

↓ indicates *OurMethod*^{2} is statistically significantly (p value < 0.05 with two-tailed t test) better than this method. The *bold face* indicates the best performance

*TUTA*1 mainly utilizes query-based features. (2) Our methods define ten similarity functions while *TUTA*1 only defines four. (3) *OurMethod*^{2} learns $\boldsymbol{\theta}$ from data for the integration of the 10 similarity functions while *TUTA*1 assigns the same weight to their similarity functions. Besides, because *CONTENT-BASED* uses the same clustering approach as *OurMethod*, the comparison with *CONTENT-BASED* indicates that the proposed *SICC* and *SISR* features are actually useful.

The intent ranking result in terms of *D#-nDCG* in Table 2 indicates that both *OurMethod*^{1} and *OurMethod*^{2} surpass *TUTA*1. This is because *TUTA*1 only considers frequency of words to model the popularity of each intent, while we also consider the intent quality and the intent relevance with respect to the query. Word frequency of intents only reflects the popularity of the intent to some extent. However, a popular intent might not be accurately clustered and is not necessarily very relevant to the given query. In another word, a popular but non-relevant or not well clustered intent should not be ranked in the top results. This issue is ignored by *TUTA*1.

Table 4 Comparison with the official result of NTCIR-11 IMine task

Participants	Intent mining (*Accuracy*)	Intents ranking (*D#-nDCG*)	Sub-intents ranking (*D#-nDCG*)	Overall performance (*H-measure*)
KLE-S-C-1A	0.5306	0.5666↓	0.6360	0.3190
KLE-S-C-2A	0.5413	0.5736	0.6339↑	0.3268
KLE-S-C-3A	0.5072↓	0.4817↓	**0.6718↑**	0.2925↓
KLE-S-C-4A	0.5148↓	0.4986↓	0.6640↑	0.2993↓
THUSAM-S-C-1A	0.5527	0.5537↓	0.4634↓	0.2811↓
THUSAM-S-C-2A	0.4347↓	0.4498↓	0.4633↓	0.1985↓
THUSAM-S-C-3A	0.3284↓	0.3744↓	0.3981↓	0.1268↓
THUSAM-S-C-4A	0.3284↓	0.3744↓	0.3993↓	0.1270↓
THUSAM-S-C-5A	0.4287↓	0.5040↓	0.4626↓	0.2072↓
CNU-S-C-1A	0.5353	0.5867	0.2045↓	0.2118↓
CNU-S-C-2A	**0.5789**	0.5569↓	0.1932↓	0.2171↓
CNU-S-C-3A	0.5086↓	0.4708↓	0.1626↓	0.1611↓
CNU-S-C-4A	0.4611↓	0.6073	0.1910↓	0.1840↓
FRDC-S-C-1A	0.2931↓	**0.7191↑**	0.3110↓	0.1510↓
FRDC-S-C-2A	0.3257↓	0.5045↓	0.2381↓	0.1209↓
FRDC-S-C-3A	0.2897↓	**0.7191↑**	0.3214↓	0.1507↓
FRDC-S-C-4A	0.5436	0.4782↓	0.2715↓	0.2038↓
FRDC-S-C-5A	0.5377	0.5004↓	0.3139↓	0.2189↓
TUTA1-S-C-1A	0.2419↓	0.3242↓	0.4391↓	0.0923↓
OurMethod^{1}	0.5260↓	0.5761	0.5371↓	0.2928↓
OurMethod^{2}	0.5681	0.6344	0.5716	**0.3426**

The official detailed results of each team for each query are available online[2]

↓ indicates *OurMethod*^{2} is statistically significantly (p value < 0.05 with two-tailed t test) better than this method

↑ indicates this method is statistically significantly (p value < 0.05 with two-tailed t test) better than *OurMethod*^{2}. The *bold face* indicates the best performance and the *underline* indicates the second best performance

The sub-intent ranking results are shown in Table 3. $TUTA1$ quantifies the quality of the list with the top k sub-intents R using definitions inspired by the metric *D#-nDCG*, i.e.

$$qual(R) = \beta N(R)/\mathcal{I}_q + (1-\beta)\sum_{r=1}^{|R|}(pop(I(SI_i))/\log(r+1)) \tag{27}$$

where $N(R)$ denotes the number of distinct clusters (intents) to which the current sub-intents R belong. $N(R)/\mathcal{I}_q$ is used to approximate *I-rec* in *D#-nDCG*. $I(SI_i)$ is the intent that the sub-intent SI_i belongs to. $pop(I(SI_i))$ is the metric used to rank intents as talked above, which is estimated using word frequency of the intent. $\sum_{r=1}^{|R|}(pop(I(SI_i))/\log(r+1))$ aims at ranking sub-intents indicating major intents in higher positions, which is used to approximate *D-nDCG* in *D#-nDCG*.

Both our methods outperform $TUTA1$ for three reasons. First, we achieve better intent mining results, which is fundamental for sub-intent ranking both in $TUTA1$ and our work. Second, $TUTA1$ simply defines the gain value of a sub-intent at rank r as the importance of its intent, i.e. $popularity(I(SI_i))$, which is not appropriate because even in the perfect clustering results, the sub-intents in the same cluster are not necessarily the same important. Third, redundancy is not considered in *TUTA*1 which results in many redundant sub-intents. E.g., for the two sub-intents “Prophet movie” and “Movie about prophet”, our method will only rank one of them at the top and filter out the other one. However, *TUTA*1 ranks both of them at the top. Finally, the learned $\boldsymbol{\theta}$ helps improve the results significantly, which confirms our arguments in Sect. 4.4. As for the ranking, we also compare our sub-intent ranking approach with *IA-Select* and *PM2*. The results indicate that our approach outperforms *IA-Select* and *PM2*, which means that our proposed sub-intent ranking algorithm is effective.

The comparison of our methods with NTCIR-11 IMine official results is shown in Table 4. The *bold face* indicates the best performance for each evaluation measure and *underline* indicates the second best performance for each evaluation measure. As we can see, the performance of our intent mining approach is only a little worse than *CNU-S-C* and our intent ranking approach is only a little worse than *FRDC-S-C* and *KLE-S-C*.[20] However, our approaches achieve the best performance in terms of *H-measure* with a 1.58 % improvement compared with the second best approach.

6.3 A case study of intents drift understanding

Unfortunately, no existing dataset and evaluation metrics are fit for measuring the effects of intent drift in intent and sub-intent ranking, we make a case study under this circumstances (Berberich and Bedathur 2013; Hu et al. 2012; Jones and Diaz 2007). Figure 6 shows the intents evolution over a timeline of the Chinese query “先知” mined by our approach. The vertical axis shows the relative popularity of four intents at that time computed with the Function 13. Table 5 shows the top five sub-intents at different time units. The results in Fig. 6 are consistent with actual facts and reflect users’ intents drift. Two most popular intents “Game” and “Movie” are more popular than the other two intents over almost all the time. Before the year 2010, the movie “Knowing” starring Nicolas Cage was the most popular intent. Then, “Dota Game” became the most popular intent which is reasonable because “Dota” became popular and many people searched the information about the “Prophet Hero (an actor in the Dota game)”. The results in Table 5 are satisfactory since the ranking changes with time to satisfy the evolving intents. Before the famous movie “Knowing” was on in 2009, “Prophet electronic dog” is ranked as the first. When the movie was popular in 2009, “Movies about prophet” is ranked as the first. After “Dota” became popular, “Playing tips of Dota prophet” is ranked as the first. Besides, the top results are diversified with different intents.

7 Conclusions and future work

In this paper, we have studied the problem of query intent mining and ranking. We implemented our approaches and baseline methods, and experimentally verified that significant improvements were achieved by our approach in terms of three popular evaluation metrics. We also demonstrated how our work helps understand user intents drift through an interesting case study.

[20] Approach details and experiment results details are not available, so we cannot carry out more experiments, analysis as well as significant test.

Table 5 Instance of Sub-intents Diversification Corresponding to Fig. 6

Time span Δt	Sub-intents Ranking
Jan.1 - Dec.31 2008	先知电子狗(Prophet electronic dog):I_4, 耶稣 (Jesus Christ):I_3, 先知攻略(Dota prophet playing tips):I_2, 先知付费影视(Prophet Films with costs):I_1, 流浪剑客(Rogue Knight, An actor in Dota):I_2
Jan.1 - Dec.31 2009	先知电影(Movies about prophets):I_1, 先知电子狗(Prophet electronic dog):I_4, 先知电影剧情(Prophet movie plot):I_1, 先知出装(Playing tips of Dota prophet):I_2, 地球毁灭密码(Chinese name of the movie “Knowing”):I_1
Jan.1 - Dec.31 2010	先知出装(Playing tips of Dota prophet):I_2, Dota 先知(Dota prophet, An actor in Dota):I_2, 先知英雄智力(Dota prophet intelligence, an attribute of the actor):I_2, 先知电影(Movies about prophets):I_1, 英雄联盟虚空先知(Prophet Void, an actor in the game League of Legends):I_2
Jan.1 - Dec.31 2011	先知出装(Playing tips of Dota prophet):I_2, Dota 先知(Dota prophet, An actor in Dota):I_2, 先知英雄力量(Dota prophet power, an attribute of the actor):I_2, 先知电影(Movies about prophets):I_1, 英雄联盟虚空先知(Prophet Void, an actor in the game League of Legends):I_2
Jan.1 - Dec.31 2012	先知出装(Playing tips of Dota prophet):I_2, Dota 先知(Dota prophet, An actor in Dota):I_2, 先知英雄智力(Dota prophet intelligence, an attribute of the actor):I_2, 先知电影(Movies about prophets):I_1, 流浪剑客(Rogue Knight, An actor in Dota):I_2
Jan.1 - Dec.31 2013	先知出装(Playing tips of Dota prophet):I_2, 先知电影(Movies about prophets):I_1, 征途电子狗(ZT electronic dog):I_4, 先知电影制作人(Prophet movie producer):I_1, 佳明电子狗(Garmin electronic dog):I_4
Full Time	先知出装(Playing tips of Dota prophet):I_2, 先知电影(Movies about prophets):I_1, Dota 先知(Dota prophet, An actor in Dota):I_2, 先知电子狗(Prophet electronic dog):I_4, 耶稣 (Jesus Christ):I_3

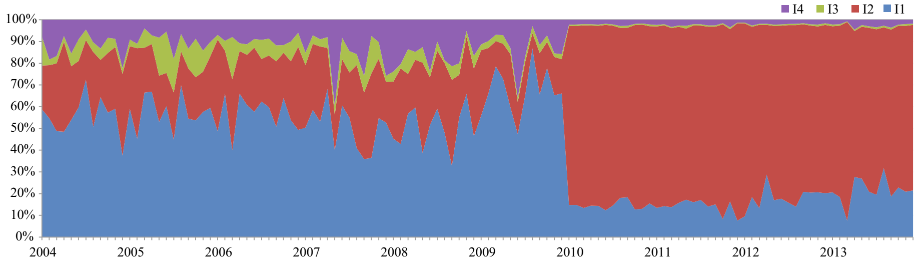


Fig. 6 The drift of four intents, I1–I4, behind the query “Prophet” over the time from 2004 to 2014. I1–I4 are “Movie”, “Game”, “Prophets in Religion” and “Prophet Electronic Dog” respectively. The *horizontal axis* represents the time, and the *vertical axis* denotes the relative popularity of the four intents computed with Function 13

There are several issues we want to further explore to enhance our current work. First, we plan to investigate the use of other similarity functions to further improve the accuracy. Second, we would like to study how to integrate the similarity functions nonlinearly. Finally, we also plan to consider more factors when ranking the intents and sub-intents to further improve the ranking results.

Acknowledgments This work is supported by the Natural Science Foundation of China (61272240, 61103151, 71402083), the Doctoral Fund of Ministry of Education of China (20110131110028), the Academy of Finland (268078), the Natural Science foundation of Shandong province (ZR2012FM037), the Excellent Middle-Aged and Youth Scientists of Shandong Province (BS2012DX017) and the Fundamental Research Funds of Shandong University.

References

Agrawal, R., Gollapudi, S., Halverson, A., & Ieong, S. (2009). Diversifying search results. In *Proceedings of the second ACM international conference on web search and data mining*, ACM, New York, NY, USA, WSDM ’09 (pp. 5–14).

Aiello, L. M., Donato, D., Ozertem, U., & Menczer, F. (2011). Behavior-driven clustering of queries into topics. In *Proceedings of the 20th ACM international conference on information and knowledge management, ACM, CIKM'11* (pp. 1373–1382).

Beeferman, D., & Berger, A. (2000). Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining, ACM* (pp. 407–416).

Beg, M. S., & Ahmad, N. (2007). Web search enhancement by mining user actions. *Information Sciences*, *177*(23), 5203–5218.

Berberich, K., & Bedathur, S. (2013). Temporal diversification of search results. In *Proceedings of the SIGIR 2013 workshop on time-aware information access*.

Biancalana, C., Gasparetti, F., Micarelli, A., & Sansonetti, G. (2013). Social semantic query expansion. *ACM Transactions on Intelligent Systems and Technology*, *4*(4), 60:1–60:43.

Calamai, P. H., & Moré, J. J. (1987). Projected gradient methods for linearly constrained problems. *Mathematical Programming*, *39*(1), 93–116.

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 *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, ACM* (pp. 875–883).

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, *46*(1–3), 131–159.

Chen, H., & Dumais, S. (2000). Bringing order to the web: Automatically categorizing search results. In *Proceedings of the SIGCHI conference on human factors in computing systems, ACM, SIGCHI'00* (pp. 145–152).

Cobos, C., Muñoz-Collazos, H., Urbano-Muñoz, R., Mendoza, M., León, E., & Herrera-Viedma, E. (2014). Clustering of web search results based on the cuckoo search algorithm and balanced Bayesian information criterion. *Information Sciences*, *281*, 248–264.

Cornuejols, G., Fisher, M. L., & Nemhauser, G. L. (1977). Location of bank accounts to optimize float. *Management Science*, *23*(8), 789–810.

Craswell, N., & Szummer, M. (2007). Random walks on the click graph. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, ACM, New York, NY, USA, SIGIR '07* (pp. 239–246).

Dang, V., & Croft, B. W. (2013). Term level search result diversification. In *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, ACM* (pp. 603–612).

Dang, V., & Croft, W. B. (2012). Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, ACM, New York, NY, USA, SIGIR '12* (pp. 65–74).

Dang, V., Xue, X., & Croft, W. B. (2011). Inferring query aspects from reformulations using clustering. In *Proceedings of the 20th ACM international conference on information and knowledge management, ACM, CIKM'11* (pp. 2117–2120).

Dou, Z, Hu, S., Chen, K., Song, R., Wen, J. R. (2011a). Multi-dimensional search result diversification. In *Proceedings of the fourth ACM international conference on web search and data mining, ACM, WSDM'11* (pp. 475–484).

Dou, Z., Hu, S., Luo, Y., Song, R., & Wen, J. R. (2011b). Finding dimensions for queries. In *Proceedings of the 20th ACM international conference on information and knowledge management, ACM, CIKM'11* (pp. 1311–1320).

Erkan, G. (2006). Language model-based document clustering using random walks. In *Proceedings of the main conference on human language technology conference of the North American chapter of the Association of Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA* (pp. 479–486).

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*(5814), 972–976.

Fujita, S., Machinaga, K., & Dupret, G. (2010). Click-graph modeling for facet attribute estimation of web search queries. In *Adaptivity, personalization and fusion of heterogeneous information, RIAO'10* (pp. 190–197).

Goyal, A., Lu, W., & Lakshmanan, L. V. (2011). Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on world wide web, ACM, WWW'11* (pp. 47–48).

Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., & Zheng, Q. (2012). Mining query subtopics from search log data. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, ACM* (pp. 305–314).

Jiang, X., & Tan, A. H. (2009). Learning and inferencing in user ontology for personalized semantic web search. *Information Sciences*, *179*(16), 2794–2808.

Jones, R., & Diaz, F. (2007). Temporal profiles of queries. *ACM Transactions on Information Systems*, *25*(3), 14:1–14:31.

Kazantseva, A., & Szpakowicz, S. (2014). Hierarchical topical segmentation with affinity propagation. In *Proceedings of the 25th international conference on computational linguistics, COLING'14* (pp. 37–47).

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, *30*(1/2), 81–93.

Kurland, O. (2006). Inter-document similarities, language models, and ad hoc information retrieval. (Doctoral dissertation, Cornell University).

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., Van Briesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, KDD'07* (pp. 420–429).

Liao, Z., Jiang, D., Chen, E., Pei, J., Cao, H., & Li, H. (2011). Mining concept sequences from large-scale search logs for context-aware query suggestion. *ACM Transactions on Intelligent Systems and Technology*, *3*(1), 171–1740.

Liu, Q., & Li, S. (2002). Word similarity computing based on hownet. *Computational Linguistics and Chinese Language Processing*, *7*(2), 59–76. (in Chinese).

Liu, Y., Miao, J., Zhang, M., Ma, S., & Ru, L. (2011). How do users describe their information need: Query recommendation based on snippet click model. *Expert Systems with Applications*, *38*(11), 13,847–13,856.

Liu, Y., Song, R., Zhang, M., Dou, Z., Yamamoto, T., Kato, M., Ohshima, H., & Zhou, K. (2014). Overview of the ntcir-11 imine task. In *Proceedings of NTCIR* (Vol. 14).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.

Marden, J. I. (1996). *Analyzing and modeling rank data*. Boca Raton: CRC Press.

Metzler, D., Dumais, S., & Meek, C. (2007). Similarity measures for short segments of text. In *Proceedings of the 29th European conference on IR research. ECIR'07* (pp. 16–27). Berlin: Springer.

Moreno, J. G., Dias, G., & Cleuziou, G. (2014). Query log driven web search results clustering. In *Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval, ACM* (pp. 777–786).

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, *14*(1), 265–294.

Pantel, P., & Lin, D. (2002). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining, ACM, New York, NY, USA, KDD '02* (pp. 613–619).

Qian, Y., Sakai, T., Ye, J., Zheng, Q., & Li, C. (2013). Dynamic query intent mining from a search log stream. In *Proceedings of the 22nd ACM international conference on conference on information and knowledge management, ACM* (pp. 1205–1208).

Radlinski, F., Szummer, M., & Craswell, N. (2010). Inferring query intent from reformulations and clicks. In *Proceedings of the 19th international conference on world wide web, ACM, WWW'10* (pp. 1171–1172).

Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the ninth machine translation summit* (pp. 315–322).

Roy, R. S., Katare, R., Ganguly, N., Laxman, S., & Choudhury, M. (2014). *Discovering and understanding word level user intent in web search queries*. Web Semantics: Science, Services and Agents on the World Wide Web.

Sadikov, E., Madhavan, J., Wang, L., & Halevy, A. (2010). Clustering query refinements by user intent. In *Proceedings of the 19th international conference on world wide web, ACM, WWW'10* (pp. 841–850).

Sakai, T., & Song, R. (2011). Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, ACM* (pp. 1043–1052).

Sakai, T., Dou, Z., Yamamoto, T., Liu, Y., Zhang, M., Kato, M. P., Song, R., & Iwata, M. (2013). Summary of the ntcir-10 intent-2 task: Subtopic mining and search result diversification. In *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, ACM* (pp. 761–764).

Sun, L., & Guo, C. (2014). Incremental affinity propagation clustering based on message passing. *IEEE Transactions on Knowledge Data Engineering*, *26*(11), 2731–2744.

Tibshirani, R. J., & Tibshirani, R. (2009). A bias correction for the minimum error rate in cross-validation. *The Annals of Applied Statistics*, *3*(2), 822–829.

Turney, P. D., Littman, M. L., Bigham, J., & Shnayder, V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. In *CoRR*.

Wang, C., Lai, J., Suen, C. Y., & Zhu, J. (2013a). Multi-exemplar affinity propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(9), 2223–2237.

Wang, Q., Qian, Y., Song, R., Dou, Z., Zhang, F., Sakai, T., et al. (2013b). Mining subtopics from text fragments for a web query. *Information Retrieval*, *16*(4), 484–503.

Wang, X., & Zhai, C. (2007). Learn from web search logs to organize search results. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, ACM* (pp. 87–94).

Wen, J., Nie, J., & Zhang, H. (2001). Clustering user queries of a search engine. In *Proceedings of the 10th international conference on world wide web, ACM, WWW'01* (pp. 162–168).

Xia, T. (2007). Study on Chinese words semantic similarity computation. *Computer Engineering*, *33*(6), 191–194. (**in Chinese**).

Xue, Y., Chen, F., Zhu, T., Wang, C., Li, Z., Liu, Y., Zhang, M., Jin, Y., & Ma, S. (2011). Thuir at ntcir-9 intent task. In *NTCIR-9 workshop meeting* (pp. 123–128).

Yu, H., & Ren, F. (2013). Tuta1 at the ntcir-10 intent task. In *Proceedings of NTCIR* (Vol. 10).

Zeng, H., He, Q., Chen, Z., Ma, W., & Ma, J. (2004). Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval, ACM* (pp. 210–217).

Zhang, X., Furtlehner, C., Germain-Renaud, C., & Sebag, M. (2014). Data stream clustering with affinity propagation. *IEEE Transactions on Knowledge Data Engineering*, *26*(7), 1644–1656.