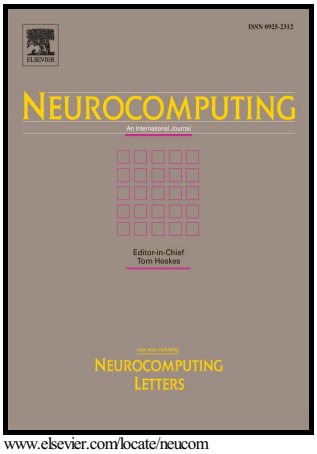


www.elsevier.com/locate/neucom

User Session Level Diverse Reranking of Search Results

Pengjie Ren, Zhumin Chen, Jun Ma, Shuaiqiang Wang, Zhiwei Zhang, Zhaochun Ren, Tinghuai Ma

PII: S0925-2312(16)30553-7
DOI: http://dx.doi.org/10.1016/j.neucom.2016.05.087
Reference: NEUCOM17163

To appear in: *Neurocomputing*

Received date: 26 February 2016
Revised date: 18 May 2016
Accepted date: 23 May 2016

Cite this article as: Pengjie Ren, Zhumin Chen, Jun Ma, Shuaiqiang Wang Zhiwei Zhang, Zhaochun Ren and Tinghuai Ma, User Session Level Diverse Reranking of Search Results, *Neurocomputing* http://dx.doi.org/10.1016/j.neucom.2016.05.087

User Session Level Diverse Reranking of Search Results

Pengjie Ren[a], Zhumin Chen[a,*], Jun Ma[a], Shuaiqiang Wang[b], Zhiwei Zhang[c], Zhaochun Ren[d], Tinghuai Ma[e]

[a]*Department of Computer Science and Technology, Shandong University, China, 250101*
[b]*Department of computer science and information systems, Jyväskylä University, Finland, 40100*
[c]*Department of Computer Science, Purdue University, The United States, IN 47907*
[d]*ISLA, Amsterdam University, The Netherlands, 1098XH*
[e]*Nanjing University of Information Science & Technology, China, 210044*

Abstract

Most Web search diversity approaches can be categorized as *Document Level Diversification* (*DocLD*), *Topic Level Diversification* (*TopicLD*) or *Term Level Diversification* (*TermLD*). *DocLD* selects the relevant documents with minimal content overlap to each other. It does not take the coverage of query subtopics into account. *TopicLD* solves this by modeling query subtopics explicitly. However, the automatic mining of query subtopics is difficult. *TermLD* tries to cover as many query topic terms as possible, which reduces the task of finding a query's subtopics into finding a set of representative topic terms. In this paper, we propose a novel *User Session Level Diversification* (*UserLD*) approach based on the observation that a query's subtopics are implicitly reflected by the search intents in different user sessions. Our approach consists of two phases: (I) *Session Graph Construction* and (II) *Diversity Reranking*. For a given query, phase (I) builds a *Session Graph* which considers relevant user sessions and preliminary retrieval results as nodes and the nodes' pairwise similarities as edge weights. Phase (II) reranks the preliminary retrieval results by minimizing a *Session Graph* based diversity loss function. Extensive experiments on two standard datasets of NACSIS Test Collections for IR (NTCIR) demonstrate the effectiveness of our approach. The advantage of our approach lies in its ability

*Corresponding author. Tel.: +86 130-7535-3300.
Email address: chenzhumin@sdu.edu.cn (Zhumin Chen)

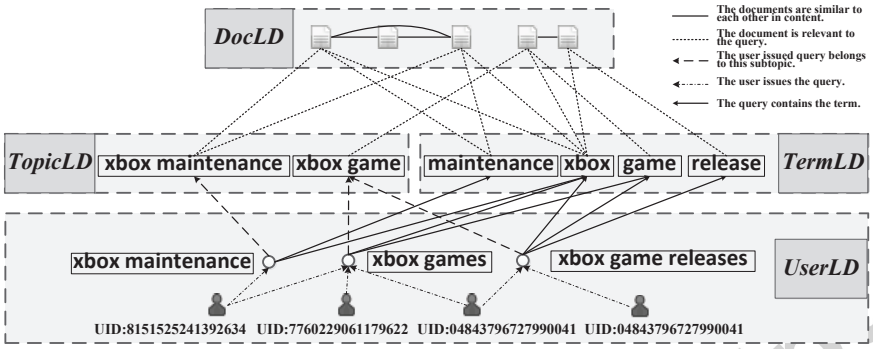
of avoiding mining the query subtopics in advance while achieving almost the same or better performances compared with previous approaches.

Keywords: Search Result Diversification, Search Result Reranking, Session Graph, User Session

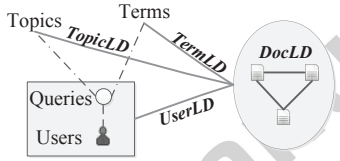
1. Introduction

Search result diversification has attracted significant attention recently as a method to improve the performance of Web retrieval systems [1, 2, 3, 4, 5]. There are at least three reasons for this. First, most queries are ambiguous and multifaceted [6, 7]. A canonical example is the query "jaguar". For this query, search engines should diversify the results because they do not know whether the query refers to the animal, the car or the software. Sometimes, even if searchers think they have offered enough information, their queries are still ambiguous. For the extended query "jaguar car", search engines still do not understand whether it represents "buying a jaguar car", "new releases of jaguar car", "price of jaguar car", and so on. Second, users' information needs are uncertain, exploratory and personalized. That is, the information needs of the same query may vary from user to user. For example, for the query "swine flu", doctors and patients may be interested in different aspects of the same topic. However, it is hard for search engines to get enough personalized information to understand users' exact intents under the current *keyword-based* search scenario. Especially when a user issues her/his first query, we know nothing (no past user behaviors) but the query. Third, content overlap exists among documents. Search engines should not return duplicate and redundant search results.

Tremendous efforts have been made on Web search result diversification [8, 9, 10], most of which can be summarized into three groups: *Document Level Diversification* (*DocLD*), *Topic Level Diversification* (*TopicLD*) and *Term Level Diversification* (*TermLD*), as shown in Figure 1. *DocLD* includes Maximal Marginal Relevance (*MMR*) [11] and its probabilistic variants [12]. They



(a) An instance of the query "xbox".



(b) Relationship.

Figure 1: Classification of search result diversification approaches.

promote diversity at document level by selecting relevant documents with the maximum contents difference. *DocLD* does not need any priori knowledge of query subtopics. However, there are no guarantees that the aspects covered by the selected documents correspond to query subtopics [13]. *TopicLD* models query subtopics explicitly and selects documents to cover as many subtopics as possible, such as *IA-Select* [13], *xQuAD* [14], *ACSL* [15] and *Proportionality Model* [16]. These approaches are generally more effective. However, as Dang and Croft [8] pointed out that, they depend heavily on a set of predefined query subtopics, the automatic mining of which is difficult and is still an ongoing research problem. Dang and Croft proposed *TermLD*, which uses a set of terms instead of subtopics of a query to promote diversity. This reduces the task of finding a query's subtopics into finding a set of representative terms.

Is there an approach that can diversify search results effectively while avoiding mining query subtopics or topic terms? In this paper, we address this problem by mining the rich human intelligence contained in query logs [17, 18]. There are two widely accepted facts. 1) User search intents behind the same query are various. The behaviors in different user sessions (issuing queries, clicking results, etc.) reflect the query's different subtopics implicitly [19, 20, 21, 22]. This means that we can promote diversity over user sessions directly in the hope that the search results can implicitly cover all subtopics of a query by covering as many relevant user sessions as possible. 2) When a user issues a query, there are usually some past user sessions with the similar search task as hers/his [23, 24], which can be reflected by similar queries, similar clicks, etc. This means that if we can find the current searcher's similar users, then a good choice is to return diversified results over the similar users. Based on above two facts, we propose a *User Session Level Diversification* (*UserLD*) approach consisting of two main phases: (I) *Session Graph Construction* and (II) *Diversity Reranking*. For a given query, phase (I) builds a *Session Graph* which considers relevant user sessions and preliminary retrieval results as nodes and the nodes' pairwise similarities as edge weights. Phase (II) reranks the preliminary retrieval results by minimizing a *Session Graph* based diversity loss function. It is worth to mention that our study in this paper focuses on the common queries that have a certain amount of query log records. Those queries are more important since they have large search volumes. Extensive experiments on two benchmark datasets in comparison with the state-of-the-art models demonstrate the effectiveness of our approach.

To sum up, the primary contributions of this paper are as follows.

- We propose a novel two-phase framework *UserLD* for search result diversification.
- Our method does not rely on the subtopic mining results while achieving better or comparable performances compared with previous methods.
- We prove that the objective function of our diversity model is *non-negative*,

monotone and *supermodular*, based on which we present an algorithm to accelerate the practical running time of the reranking phase.

The remaining sections are organized as follows. In Section 2, we discuss related work. In Section 3, we present our approach to implement *UserLD*. In Section 4, we report our experiment results. Section 5 concludes our study and discusses future work.

2. Related Work

There is a large amount of previous work on search result diversification. Existing studies usually classify search result diversification as either explicit or implicit based on whether the approach models query subtopics explicitly or not. In this paper, we group existing diversification approaches into three classes from a different perspective: *DocLD*, *TopicLD* and *TermLD*.

DocLD approaches pursue a balance between content novelty and relevance of documents. *MMR* [11] is one of the early influential work belonging to *DocLD*. *MMR* diversifies search results by comparing the overlap of documents contents and gradually selecting the next document which is relevant to the query meanwhile contains minimal similarity to previously selected documents. Its variant proposed by Zhai et al. [12] revised the framework from a probabilistic perspective. They promoted diversity by considering KL divergence of the documents' language models. Wang et al. promoted diversity by selecting documents that are different to one another in terms of vocabulary, as captured by Pearson's correlation between retrieved documents [25]. Liang et al. [26] proposed a supervised learning approach that diversifies search results by adding diversity constraints to structured SVM learning framework. Zhu et al. [27] considered the process of diversity as a sequential selection process. They first defined several diversity related features. Then, they learned a diversity ranking function by minimizing a likelihood loss of the generation probability.

TopicLD approaches model the set of query subtopics and return relevant documents to cover as many subtopics as possible. One of the state-of-art mod-

els, *IA-Select* [13], supposes users only consider the top k returned results of a search engine. *IA-Select* tries to maximize the probability that there is at least one relevant result for each subtopic within the top k results. Another model, *xQuAD* [14], explicitly accounts for the various subtopics associated to a query. It diversifies search results by estimating how well a given document satisfies each uncovered subtopic and the extent to which different subtopics are already satisfied by the results as a whole. Dou et al. proposed *ACSL* [15] which assumes that a good diversified result should cover as many subtopics as possible in multiple dimensions, and at the same time the relevance of results should be preserved. The framework can be regarded as a general form of the *xQuAD* framework, the *MMR* model, and the *IA-Select* model. Dang et al. [16] supposed that the number of results belonging to each subtopic should be proportional to the subtopic's popularity. They treated the diversification problem as finding a proportional representation process over different subtopics for the document ranking. Raman et al. [2] addressed the problem of intrinsic diversity, which has little ambiguity in intent but pursues content coverage of aspects on a certain subtopic. Their target was not a single query, but a type of complex task which spans multiple queries across one or more user search sessions. Santos et al. [28] assumed that there are various subtopics underlying a query, and that users' information needs include navigational intents and informational intents. They first learned the appropriateness of different retrieval models for each of the aspects underlying this query. Then they proposed an intent-aware search result diversification method to cover all subtopics and intents. Hong and Si [3] introduced two approaches to diversify results of resource selection in distributed information retrieval. The first approach reranks the documents based on their relevance to the query subtopics. The second approach estimates the relevance of each information source with respect to different subtopics of the query by any existing resource selection algorithm.

TermLD approaches work by diversifying search results based on a set of query topic terms. Those terms can be identified from the summarization of the preliminary documents ranking. Dang and Croft [8] proved the effectiveness

of *TermLD* and concluded that grouping those terms into subtopics provides little benefit to diversification compared to the presence of the terms themselves. This reduces the task of finding a set of query subtopics into finding a simple set of topic terms.

In this paper, we propose a new approach, namely *UserLD*. Different from *DocLD*, *TopicLD* and *TermLD*, *UserLD* promotes diversity based on past user search sessions. The intuition is that different user sessions may have different search intents. Those different search intents reflect various subtopics of a query. *UserLD* tries to maximize the coverage of user search intents and query subtopics by covering as many user sessions as possible.

3. User Session Level Search Result Diversification

3.1. Notions and Notations

Before introducing our approach, we introduce some notations and key concepts. A summary of the notions and notations is shown in Table 1. Let $u \in U$ represent a user session. A user session usually starts with a user sending a query to a search engine, receiving a list of ranked documents, then examining the snippets, clicking on the interesting ones, and spending more time reading them. Then the user modifies the query or issues a new query to start the search again. The process iterates until the user's information need is satisfied or the user abandons the search, which ends the session. Let $Q(u)$ represent all issued queries and $C(u)$ represent all clicked URLs in the session u.

User queries can be classified into four types of patterns [21]. If the query is a single phrase, usually a noun phrase, then the type is "Q". The other three types are "Q + W", "W + Q", and "Others", where "W" denotes some keywords [29, 30]. For example, "Q": Harry Potter, "Q + W": Harry Potter movie, "W + Q": download Harry Potter. According to [21], the percentages of the four types are 45.5%, 25.5%, 16.5% and 12.5% respectively. This is reasonable because users tend to add additional keywords to specify their search intents in their minds when the current results are not satisfactory [31]. Let u_c

Table 1: Summary of Notions and Notations.

U	All user sessions.
D	All documents.
$u, u' \in U$	A user session.
$d, d', d'' \in D$	A document.
$Q(u)$	The collection of issued queries in the session u.
$C(u)$	The collection of clicked URLs in the session u.
q_c	The current issued query whose results need to be diversified.
u_c	The current user session that q_c belongs to.
Q_{q_c}	The collection of "Q"-type, "Q + W"-type and "W + Q"-type queries corresponding to query q_c in the query logs.
$Q_{q_c}(u)$	The collection of "Q"-type, "Q + W"-type and "W + Q"-type queries corresponding to query q_c in the session u.
C_{q_c}	The collection of all clicked URLs in the query logs where the issued queries belong to Q_{q_c}.
$C_{q_c}(u)$	The collection of clicked URLs in the session u where the issued queries belong to Q_{q_c}.
$U_{q_c} \subseteq U$	The collection of user sessions that contain at least one query belonging to Q_{q_c}.
$D_{q_c} \subseteq D$	The preliminary results of query q_c returned by BM25.
$R_{q_c} \subseteq D_{q_c}$	The reranking results of query q_c.

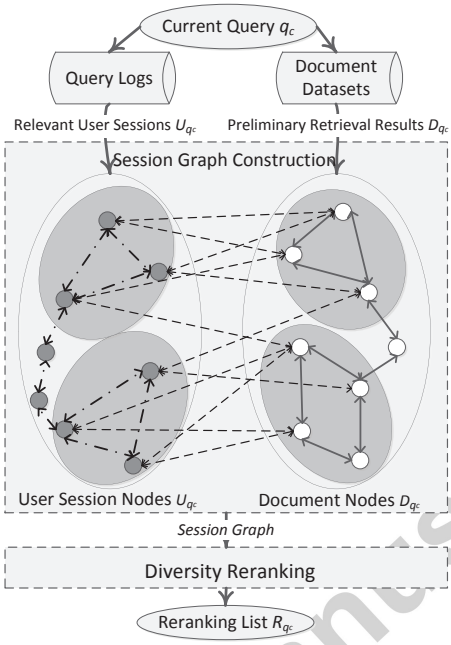


Figure 2: Process of Our Approach.

represent the current user session and q_c represent the current issued query in the session which needs to be diversified. Then we consider q_c as a "Q"-type query, and find out all "Q + W"-type and "W + Q"-type queries in the query logs. All those queries are considered as the related queries of q_c, denoted as Q_{q_c} (including q_c itself). All user sessions containing one or more queries of Q_{q_c} constitute the session set $U_{q_c} \subseteq U$.

Our approach consists of two main phases: *Session Graph Construction* and *Diversity Reranking*, as shown in Figure 2. For a given query q_c, we first retrieve all user sessions which contain the query string (i.e., U_{q_c}) and use the BM25 model with Lucene implement to generate the top 100 preliminary document results (i.e., D_{q_c}). Then, we consider U_{q_c} and D_{q_c} as nodes and the nodes' pairwise similarities as edge weights to build the *Session Graph*. Finally, we rerank D_{q_c} by minimizing a *Session Graph* based diversity loss function.

3.2. Session Graph Construction

The *Session Graph* $G(u_c, q_c)$ for the current user session u_c and current issued query q_c can be formalized as a four-tuples $(V, E, W(U_{q_c}), P(E))$ [32]. $V = \{U_{q_c} \cup D_{q_c}\}$ is the collection of nodes. There are two kinds of nodes: user session nodes U_{q_c} and document nodes D_{q_c}. Each session node $u \in U_{q_c}$ is associated with a weight value $w(u) \in W(U_{q_c})$ reflecting the importance of u. We consider several aspects to model $w(u)$ in the next subsession. $E = \{e_{(u \leftrightarrow u')}, e_{(d \leftrightarrow d')}, e_{(d \leftrightarrow u)} | u \neq u' \in U_{q_c}; d \neq d' \in D_{q_c}\}$ is the collection of edges. Each edge is associated with a probability weight value $P(e) \in P(E)$ reflecting the similarity of the two nodes [33]. There are two key problems in building the *Session Graph*. 1) How to define the node weight $w(u)$? 2) How to define the edge weight $P(e)$ and compute the pairwise edge weights efficiently? We investigate the two problems respectively.

3.2.1. Node Weight $w(u)$

$w(u)$ is defined as follows:

$$w(u) = imp(u) \cdot sim(u_c, u); \tag{1}$$

Formula 1 contains two parts. The first part (i.e., $imp(u)$) is the priori importance of session u, which is estimated based on two aspects: issuing a popular query, clicking a popular URL. The two aspects are balanced with a parameter α.

$$\begin{aligned} imp(u) &= \alpha pop_{query}(u) + (1-\alpha) pop_{click}(u) \\ &= \alpha \frac{\max\limits_{q' \in Q_{q_c}(u)} \ln vol(q')}{\max\limits_{q'' \in Q_{q_c}} \ln vol(q'')} + (1-\alpha) \frac{\max\limits_{url' \in C_{q_c}(u)} \ln cli(url')}{\max\limits_{url'' \in C_{q_c}} \ln cli(url'')}; \end{aligned} \tag{2}$$

where $Q_{q_c}(u)$ and $C_{q_c}(u)$ are queries and clicked URLs in session u corresponding to query q_c; C_{q_c} is the collection of clicked URLs corresponding to Q_{q_c}; $vol(q)$ is the search volume of q; $cli(url)$ is the click volume of url. $\ln(\cdot)$ form is adopted to reduce the exponential growth of search/click volumes. $imp(u)$ is computed offline.

The second part (i.e., $sim(u_c, u)$) is the similarity of session u to the current session u_c.

$$sim(u_c, u) = \frac{\sum_{i=1}^{4} s^i(u_c, u)}{4}; \tag{3}$$

where $s^i(u_c, u)$ is a sub-similarity function. There are four sub-similarity functions in total. Although $sim(u_c, u)$ needs to be computed online, however it can be computed efficiently in parallel for each $u \in U_{q_c}$. Given any two user sessions u and u', we propose two *Query Similarities* $s^1(\cdot)$, $s^2(\cdot)$ and two *Click Similarities* $s^3(\cdot)$, $s^4(\cdot)$ to evaluate their similarity.

Query Similarity. Query similarities measure the similarity of user search intents [34, 35]. We define two query similarity functions in this paper.

The first similarity function describes the term match between the queries.

$$\begin{aligned} s^1(u_c, u) &= \frac{1}{|Q(u)|} \sum_{q_j \in Q(u)} \max_{q_i \in Q(u_c)} \frac{tms(q_i, q_j) + tms(q_j, q_i)}{2}; \\ tms(q_i, q_j) &= \frac{1}{|q_i|} \sum_{m=1}^{|q_i|} \frac{|q_j| - \min\{|q_j|, |m-n| | n \in Pos(q_j, q_i^m)\}}{|q_j|}; \end{aligned} \tag{4}$$

where q_i^m represents the term at position m of query q_i, $Pos(q_j, q_i^m)$ is the set of all positions of query q_j where the term is q_i^m. $tms(q_i, q_j)$ equals 1 if q_i is the same with q_j exactly and 0 if no terms overlap exists. Otherwise, $tms(q_i, q_j)$ is between 0 and 1 according to term sequence consistency. For each $q_j \in Q(u)$, we find its most similar $q_i \in Q(u_c)$ according to $tms(q_i, q_j)$ and $tms(q_j, q_i)$. Then we average the similarities as $s^1(u_c, u)$.

While the queries in two user sessions may not have direct term overlap, they may be similar semantically. To address this, the second function measures the semantic similarity of two queries:

$$\begin{aligned} s^2(u_c, u) &= \frac{1}{|Q(u)|} \sum_{q_j \in Q(u)} \max_{q_i \in Q(u_c)} \frac{P(q_i|q_j) + P(q_j|q_i)}{2}; \\ P(q_i|q_j) &= \frac{1}{|q_i|} \sum_{m=1}^{|q_i|} \max_{q_j^n} P(q_i^m | q_j^n); \end{aligned} \tag{5}$$

In this paper, the word translation probability $P(q_i^m|q_j^n)$ is estimated offline based on the queries derived from the search logs [36, 34].

$$P(q_i^m|q_j^n) = \frac{TF(q_i^m, q_j^n)}{TF(q_j^n)}; \tag{6}$$

where $TF(q_i^m, q_j^n)$ is the co-occurrence frequency of the two words in user issued queries; $TF(q_j^n)$ is term frequency of q_j^n in user issued queries. $P(q_i^m|q_j^n)$ measures the co-occurrence probability of two words. So $s^2(u_c, u)$ reflects the semantic similarity of u_c and u.

Click Similarity. The clicked URLs provide a different source of information about users' search intent besides queries [37, 38]. We define two click similarity functions in this paper.

The first is based on Jaccard similarity.

$$s^3(u_c, u) = \frac{|C(u) \bigcap C(u_c)|}{|C(u) \bigcup C(u_c)|}; \tag{7}$$

where $C(u)$ is the set of clicked URLs in the session u. $s^3(u_c, u)$ measures the direct clicks overlap between u_c and u.

While the clicked results in two user sessions may not have direct overlap in terms of clicked URLs, they may belong to the same topic. To address this, we define the second similarity based on Cosine similarity.

$$\begin{aligned} s^4(u_c, u) &= \frac{1}{|C(u)|} \sum_{url' \in C(u)} \max_{url \in C(u_c)} s(url, url'); \\ s(url, url') &= \frac{\vec{W}_{url'} \cdot \vec{W}_{url}}{\| \vec{W}_{url'} \| \cdot \| \vec{W}_{url} \|}; \end{aligned} \tag{8}$$

where url and url' represent clicked URLs, $\vec{W}_{url}$ denotes the word vector of all issued queries when users click the URL. $s^4(u_c, u)$ is based on the intuition that the issued queries are good topic indicators of clicked URLs.

3.2.2. Edge Weight $P(e)$

Each document node is represented as a $TF\text{-}IDF$ vector, $TF\text{-}IDF(d)$. Each user session node is represented as a $TF\text{-}IDF$ vector of the issued queries and

clicked documents, i.e.,

$$TF\text{-}IDF(u) = \beta \frac{\sum_{q \in Q(u)} TF\text{-}IDF(q)}{|Q(u)|} + (1-\beta) \frac{\sum_{d \in C(u)} TF\text{-}IDF(d)}{|C(u)|}; \quad (9)$$

Then the edge weight probability $P(e)$ is estimated as follows. β is the hyper parameter which will be detailed in the experiments.

$$\begin{aligned} P(e_{(u \leftrightarrow u')}) &\approx Cosine(TF\text{-}IDF(u), TF\text{-}IDF(u')); \\ P(e_{(d \leftrightarrow d')}) &\approx Cosine(TF\text{-}IDF(d), TF\text{-}IDF(d')); \\ P(e_{(d \leftrightarrow u)}) &\approx Cosine(TF\text{-}IDF(u), TF\text{-}IDF(d)); \end{aligned} \quad (10)$$

We evaluate $P(e)$ with a Cosine similarity for two reasons. First, the range of Cosine similarity is $[0, 1]$ which satisfies the range of probability. Second, the random projection method of LSH [39] (also known as *SimHash*) can be used to approximately and efficiently find the n-nearest user sessions or documents. A summary of the *Session Graph* construction is shown in Algorithm 1. The computational complexities of the offline and online phase are $O(|D| + |U|)$ and $O(|U_{q_c}||NerU| + |D_{q_c}||NerU||NerD|)$. $|D|$ and $|U|$ are large but $|U_{q_c}|$, $|D_{q_c}|$, $|NerU|$, $|NerD|$ are small, so most time consuming work is done offline. Also the size of $|U_{q_c}|$, $|D_{q_c}|$, $|NerU|$, $|NerD|$ does not change as the increase of the dataset. As a result, the online phase is efficient.

3.3. Diversity Reranking

Given the *Session Graph* $G(u_c, q_c)$, we define the diversity loss function $L(R, G(u_c, q_c))$ based on $G(u_c, q_c)$ as:

$$\begin{aligned} L(R, G(u_c, q_c)) &= \sum_{u \in U_{q_c}} w(u) \overline{P}_{R \Rightarrow u}; \\ \overline{P}_{R \Rightarrow u} &= \prod_{path \in path(R \Rightarrow u)} (1 - \prod_{e \in path} P(e)); \end{aligned} \quad (11)$$

$path(R \Rightarrow u) = \bigcup_{d \in R} path(d \Rightarrow u)$ where $path(d \Rightarrow u)$ is the collection of all simple paths (no cycle) from d to u in $G(u_c, q_c)$. The loss function contains two

Algorithm 1: *Session Graph* Construction.

Input:

Current user search session, u_c; Current issued query that needs to be diversified, q_c; The number of nearest neighbors, n; The number of session nodes, $|U_{q_c}|$; The preliminary result set D_{q_c};

Output:

The *Session Graph*, $G(u_c, q_c)$;

Offline Phase:

Employ Lucene to index query logs and documents;
Employ *SimHash* to index user session vectors $TF\text{-}IDF(u)$ and document vectors $TF\text{-}IDF(d)$;
for each $d \in D$ (in parallel) **do**
    Search *SimHash* index to find d's n nearest user sessions $NerU(d)$ and n nearest documents $NerD(d)$;
end for
for each $u \in U$ (in parallel) **do**
    Search *SimHash* index to find u's n nearest user sessions $NerU(u)$;
end for

Online Phase:

Search Lucene index to form the user session set U_{q_c};
for each $u \in U_{q_c}$ (in parallel) **do**
    Compute u's node weight $w(u)$;
    Get u's n nearest user sessions $NerU(u)$;
    for each $u' \in NerU(u)$ (in parallel) **do**
        Compute u''s node weight $w(u')$;
        Link u and u' with edge $e_{(u \leftrightarrow u')}$ and edge weight $P(e_{(u \leftrightarrow u')})$;
    end for
end for
for each $d \in D_{q_c}$ (in parallel) **do**
    Get d's n nearest user sessions $NerU(d)$;
    for each $u' \in NerU(d)$ (in parallel) **do**
        Compute u''s node weight $w(u')$;
        Link d and u' with edge $e_{(d \leftrightarrow u')}$ and edge weight $P(e_{(d \leftrightarrow u')})$;
    end for
    Get d's n nearest documents $NerD(d)$;
    for each $d' \in NerD(d)$ (in parallel) **do**
        Link d and d' with edge $e_{(d \leftrightarrow d')}$ and edge weight $P(e_{(d \leftrightarrow d')})$;
    end for
end for
return $G(u_c, q_c)$;

parts. The first part (i.e., $w(u)$) is the importance of u. This part reflects the satisfaction to u_c if the final results R cover u. $\prod_{e \in path} P(e)$ is the probability that R covers u along the path *path* in $G(u_c, q_c)$. So the second part (i.e., $\overline{P}_{R \Rightarrow u}$) is the probability that u is not covered by the results R along any path, which corresponds to loss. We minimize $L(R, G(u_c, q_c))$ by selecting $|R|$ documents to cover as many user sessions as possible in terms of $w(u)$. Because different user sessions correspond to different query subtopics and $w(u)$ corresponds to the importance, the result R that minimizes Formula 11 actually is diversified implicitly.

The only variable in our loss function (Formula 11) is search results R. So Formula 11 is abbreviated as $L(R)$ without causing ambiguity next. A frequently-used algorithm for selecting the results R from D_{q_c} by existing approaches is *Greedy*: start with the empty list, and repeatedly add a document d that maximizes $L(R) - L(R \cup \{d\})$. However, the running time of the *Greedy Algorithm* is unacceptable for larger $G(u_c, q_c)$ graphs, because for each step, we need to calculate $L(R \cup \{d\})$ for every remaining document node $d \in D_{q_c} \setminus R$. We take three measures to speed up Greedy.

(1) Lazy Greedy. We can prove that Formula 11 is *non-negative*, *monotone* and *supermodular*.

Theorem 1. $L(R, G(u_c, q_c))$ is *non-negative, monotone* and *supermodular.* i.e. $L(R)$ satisfies the following properties:

1) *non-negative*: $L(R) \geq 0, \forall R \subseteq D_{q_c}$;

2) *monotone*: $L(R^i) \geq L(R^{i+1}), \forall R^i \subseteq R^{i+1}$;

3) *supermodular*: $\triangle L_d(R^i) = L(R^i) - L(R^i \cup \{d\}) \geq \triangle L_d(R^{i+1}) = L(R^{i+1}) - L(R^{i+1} \cup \{d\}), \forall R^i \subseteq R^{i+1}$;

The proof details of Theorem 1 are shown in appendix. Based on Theorem 1, the practical running time of Greedy can be alleviated by Lazy Greedy (or Accelerated Greedy) [40]. The key idea of applying Lazy Greedy to our problem is that, according to Theorem 1, as the results R grows, the increments $\triangle L_d(R)$ will never increase. Assume R^i is the results after adding the

ith document. When selecting the next document d, instead of recomputing $\triangle L_d(R^i) = L(R^i) - L(R^i \cup \{d\})$ for every remaining document node $d \in D_{q_c} \setminus R^i$, we maintain a table of document nodes sorted on $\triangle L_d(R)$ in decreasing order. $\triangle L_d(R^i)$ is re-evaluated only for the top document node at a time. If the node remains at the top, we add it to R^i as the next selected document without re-computing $\triangle L_d(R^i)$ for the other nodes in the table. Otherwise d is re-added to the table and the table is resorted. The reason is simple. Assume the top node in the table is d'. d'' is any other node in the table ($d'' \neq d'$). Also assume d' remains at the top after evaluating $\triangle L_{d'}(R^i)$, i.e. $\triangle L_{d'}(R^i) \geq \triangle L_{d''}(R), R \subseteq R^i$. Since $L(R)$ is supermodular, we further have $\triangle L_{d''}(R) \geq \triangle L_{d''}(R^i)$. So we can conclude that $\triangle L_{d'}(R^i) \geq \triangle L_{d''}(R^i)$ without computing $\triangle L_{d''}(R^i)$ for each remaining node d''.

(2) Avoid Redundant Computation. Another way to speed up the optimization process is avoiding computing $\triangle L_d(R)$ from scratch for each iteration of Lazy Greedy. With simple reduction, we have

$$\begin{aligned} &\triangle L_d(R, G(u_c, q_c)) \\ &= L(R, G(u_c, q_c)) - L(R \cup \{d\}, D_{q_c}, G(u_c, q_c)) \\ &= \sum_{u \in U_{q_c}} w(u) \overline{P}_{R \Rightarrow u} P_{d \Rightarrow u}; \end{aligned} \tag{12}$$

Note that $path(R \Rightarrow u) = \bigcup_{d' \in R} path(d' \Rightarrow u)$. As a result,

$$\triangle L_d(R, G(u_c, q_c)) = \sum_{u \in U_{q_c}} w(u) \left[\prod_{d' \in R} (1 - P_{d' \Rightarrow u}) \right] P_{d \Rightarrow u}; \tag{13}$$

From Formula 13 we can see that, once $P_{d \Rightarrow u}$ is computed for each $d \in D_{q_c}$ in the first iteration of Lazy Greedy, $\triangle L_d(R, G(u_c, q_c))$ does not need to be computed from scratch for all left iterations. Instead, with Formula 13 it can be computed efficiently.

(3) Monte Carlo Simulation. The remaining part is $P_{d \Rightarrow u}$. As $P_{d \Rightarrow u} = 1 - \prod_{path \in path(d \Rightarrow u)} \left(1 - \prod_{e \in path} P(e)\right)$, so the time complexity of computing $P_{d \Rightarrow u}$ is equal to finding all simple paths from d to u, which is efficient enough on small sparse graphs. However, on large dense graphs, it is still time consum-

ing. Fortunately, this problem can be alleviated by using *Monte Carlo* (*MC*) simulation. Specifically, we generate a uniformly random value $P_{ran} \in [0, 1]$ to decide if a node can activate the node on the other side of an edge (activated if $P_{ran} > P(e)$). We call this "activating step". One *MC* iteration for $P_{d \Rightarrow u}$ goes by regarding d as the initial node and repeating activating step until no new activated nodes. If u is activated in this iteration, we increase the count $N(d \Rightarrow u)$ and add it to $\sigma(d)$. $\sigma(d)$ represents the activated user session nodes by d. Typically, the process repeats $N = 10,000$ times in parallel. Assume the total activated count of u is $N(d \Rightarrow u)$, then we estimate $P_{d \Rightarrow u} \approx \frac{N(d \Rightarrow u)}{N}$.

Algorithm 2 summarizes the diversity reranking algorithm. For each node d, we store a three-tuples of the form $< d.inc, d.flag, d.map >$. Here $d.inc = \triangle L_d(R)$ and $d.flag$ is the iteration number when $d.inc$ was last updated. $*.map = \{< u, P_{* \Rightarrow u} > | u \in \sigma(*)\}$, where $*$ represents a single document or a set of documents and $\sigma(*)$ represents the activated user session nodes by $*$ and $P_{* \Rightarrow u}$ is the probability that u is activated by $*$, i.e. $P_{* \Rightarrow u} \approx \frac{N(* \Rightarrow u)}{N}$ with *MC* implement. As a result, $\triangle L_d(R)$ is updated with *MC* as follows:

$$\begin{aligned} &\triangle L_d(R, G(u_c, q_c)) \\ &= \sum_{u \in U_{q_c}} w(u) \overline{P}_{R \Rightarrow u} P_{d \Rightarrow u} \\ &= \sum_{u' \in \sigma(d) \setminus \sigma(R)} w(u') P_{d \Rightarrow u'} + \sum_{u' \in \sigma(d) \cap \sigma(R)} w(u')(1 - P_{R \Rightarrow u'}) P_{d \Rightarrow u'}; \end{aligned} \tag{14}$$

The first part is the expected increment by newly activated nodes in $\sigma(d)$ and the second part is the expected increment by overlap $\sigma(d) \cap \sigma(R)$. Similarly, $(R \cup \{d\}).map$ is updated using

$$\begin{aligned} (R \cup \{d\}).map = &\{< u', P_{R \Rightarrow u'} > | u' \in \sigma(R) \setminus \sigma(d)\} \\ \cup &\{< u', P_{d \Rightarrow u'} > | u' \in \sigma(d) \setminus \sigma(R)\} \\ \cup &\{< u', 1 - (1 - P_{R \Rightarrow u'})(1 - P_{d \Rightarrow u'}) > | u' \in \sigma(R) \cap \sigma(d)\}; \end{aligned} \tag{15}$$

The most time consuming task of Algorithm 2 is the *Monte Carlo Simulation* for each $d_j \in D_{q_c}$ (line 2 to 7). In our experiments, the *Session Graph*s are built

Algorithm 2: Diversity Reranking.

Input:

The graph, $G(u_c, q_c)$;

The number of results, k;

Output:

k document nodes, R;

Initialize the maximum heap H, the list R, the map $R.map$;
for each $d_j \in D_{q_c}$ (in parallel) **do**
  Initialize the map $d_j.map$;
  Estimate $L(d_j)$ and $d_j.map = \{< u, \frac{N(d_j \Rightarrow u)}{N} > |u \in \sigma(d_j)\}$ with MC;
  Set $d_j.inc = L(d_j)$, $d_j.flag = 0$;
  $Put(H, (d_j, d_j.inc))$;
end for
while $|R| < k$ **do**
  $d = Pop(H)$;
  if $d.flag == |R|$ **then**
    Add d to R;
    Update $R.map$ using Formula 15;
  else
    Compute $\triangle L_d(R)$ using Formula 14;
    Set $d.inc = \triangle L_d(R)$, $d.flag = |R|$;
    $Put(H, (d, d.inc))$;
  end if
end while
return R;

from commercial query logs collected in one month. In these *Session Graph*s, the *Monte Carlo Simulation* for each $d_j \in D_{q_c}$ can be finished in milliseconds.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets

To evaluate the performance of *UserLD*, we used the standard dataset from INTENT-1[1] and INTENT-2[2] task. The INTENT-1 Chinese dataset contains 100 Chinese queries with more than 900 subtopics. The INTENT-2 Chinese dataset contains about 100 Chinese queries with more than 600 subtopics. The subtopic importance distribution $P(z|q)$ (where z is a subtopic of query q) is also given for each query. The corresponding query logs[3] and Web page collection[4] used in the experiment are from a Chinese commercial search engine. The query logs contain 51.4 million log records collected in a month. The Web page collection contains 135.4 million Web pages from 5.3 million Chinese Web sites, and the total uncompressed storage size is about 5.0 TBytes. We adopt this dataset because this is the only available dataset with large amount of real user search logs, to the best of our knowledge.

4.1.2. Relevance Assessments

For evaluating the performance of diversity, we have to judge the relevant level for each document-subtopic pair as ground truth. The ground truth is built using conventional pooling approach. The same interface was used for four assessors (two undergraduates and two graduates), which lets assessors view each pooled document and select a relevance grade for each subtopic: "Excellent", "Great", "Good", "Fair", "Bad". Two assessors were assigned to each query. Similar to the work in [41], we also assumed that the disagreements between

[1] http://www.thuir.org/intent/ntcir9/
[2] http://research.microsoft.com/en-us/projects/intent/
[3] http://www.sogou.com/labs/dl/q-e.html
[4] http://www.sogou.com/labs/dl/t-e.html

Table 2: Summary of Diversity Models in Experiments.

Category	Symbol	Explanations
DocLD	*MMR*	*MMR* proposed in [11].
TopicLD	*IA-Select(TopicLD,AVG)*	*IA-Select* [13] with given subtopics and uniformly distributed subtopic importance.
	IA-Select(TopicLD,GT)	*IA-Select* [13] with given subtopics and subtopic importance distribution in ground truth.
	xQuAD(TopicLD,AVG)	*xQuAD* [14] with given subtopics and uniformly distributed subtopic importance.
	xQuAD(TopicLD,GT)	*xQuAD* [14] with given subtopics and subtopic importance distribution in ground truth.
TermLD	*IA-Select(TermLD)*	*IA-Select* implemented with term level diversification approach proposed in [8].
	xQuAD(TermLD)	*xQuAD* implemented with term level diversification approach proposed in [8].
UserLD	*UserLD*	User session level search result diversification approach proposed in this paper.

the two assessors are negligible. The relevance grades were aggregated to form a five-point relevance scale, from L0 ("Bad") to L4 ("Excellent").

4.1.3. Baseline Diversity Models

We use seven classic models in diversity literature as baselines as shown in Table 2. The first is a *DocLD* approach, i.e. *MMR* [11]. The *TopicLD* approaches include *IA-Select* [13] and *xQuAD* [14]. The *TermLD* approaches include *IA-Select* and *xQuAD* model implemented with term level diversification approach proposed in [8].

4.1.4. Experimental Tools and Parameter Settings

The query logs and documents are indexed with Lucene (Version 4.3.0)[5]. The session vectors ($TF\text{-}IDF(u)$) and document vectors ($TF\text{-}IDF(d)$) are indexed with TarsosLSH (Version 0.7)[6]. We use Lucene BM25 to generate the top 100 preliminary results (as D_{q_c}) for both our approach and the baselines. We use all relevant user sessions in the one-month query logs to build the *Session Graph*s, i.e., U_{q_c} is the set of user sessions where at least one of the issued queries contains the current query q_c. The parameters of our approach (α, β) and the baselines are tried from 0.1 to 1.0 with a step of 0.1. The best run for each approach is adopted for comparison. All experiments were carried out on a server with 64 cpu cores (Intel(R) Xeon(R) E7-4820 @ 2.00GHz) and 132G memory.

4.1.5. Evaluation Metrics

The experiment results are measured with five standard metrics that have been widely used in evaluation of Web search result diversification: *I-rec*, $nDCG\text{-}IA$, $nERR\text{-}IA$, *D#-nDCG*, *D#-Q*. *I-rec* indicates how many subtopics are covered by the search results. The *IA* metrics were proposed by Agrawal et al. [13] as a simple methodology for evaluating diversified search results. Take $nDCG\text{-}IA$ as an example, it is computed as:

$$nDCG\text{-}IA@k = \sum_{z} P(z|q) nDCG_z@k; \tag{16}$$

where z is a subtopic of query q; $P(z|q)$ is the subtopic importance distribution; $nDCG_z$ is $nDCG$ for a particular subtopic z. $nERR\text{-}IA$ can be computed similarly.

In order to solve the undernormalisation problem of *IA* metrics, Sakai et al. [42] proposed *D#-metrics*, which is computed as:

$$D\#\text{-}metric@k = \gamma I\text{-}rec@k + (1-\gamma) D\text{-}metric@k; \tag{17}$$

[5] https://lucene.apache.org/. Query log fields: Session ID, Issued Query, Clicked Document ID. Document fields: Document ID, Content.

[6] https://github.com/JorenSix/TarsosLSH

where γ is a hyper parameter which will be detailed later. *D-metric* is computed by replacing the raw gain $g(r)$ of cumulative-gain-based metrics such as *nDCG* and *Q-measure* with the global gain:

$$GG(r) = \sum_{z} P(z|q) g_z(r); \tag{18}$$

where $g_z(r)$ is the gain value of document at rank r to subtopic z.

Most diversification mechanisms are evaluated using only diversity measures. However, the diversity may be achieved at a cost of relevance. Therefore, in addition to above diversity measures, we also evaluate our results using four standard relevance-based metrics for Web retrieval: *nDCG*, *nERR*, *P@k*, and *Q-measure*. All of these metrics are computed using NTCIREVAL toolkit[7].

4.2. Evaluation of Result Diversity

4.2.1. IA metrics Results

The performance quantified by *IA* metrics is depicted in Figure 3. The *IA* metrics evaluate the diversity of the ranking results by computing the weighted sum of per-subtopic relevance, which forces a tradeoff between selecting documents with higher relevance scores and those that cover additional subtopics by taking into account the importance distribution of subtopics. For both INTENT-1 and INTENT-2 datasets, the results are almost identical. At all rank thresholds (top k=1 to 20) and all *IA* metrics we evaluated, the orderings produced by *UserLD* are better than all other baselines. Specially, the improvement of *nDCG-IA* is significant. Furthermore, as higher thresholds are considered, the outperformance of *UserLD* increases. However, since *IA* metrics evaluate the diversity performance by considering relevance and subtopic coverage as a whole, we do not know the improvement comes from better relevance or better subtopic coverage. So we further evaluate the results with *I-rec*.

[7] http://research.nii.ac.jp/ntcir/tools/ntcireval-en.html

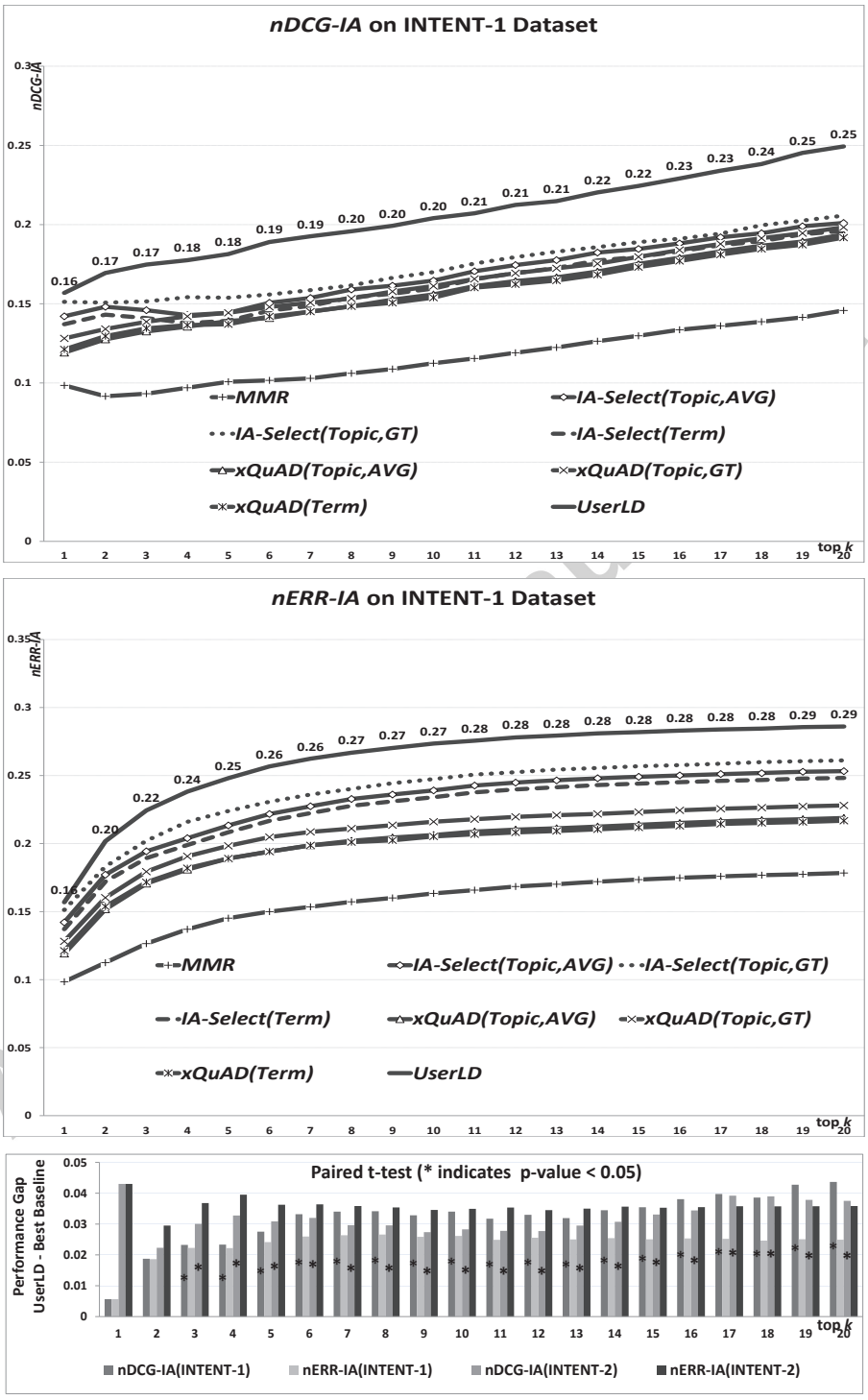


Figure 3: *IA* metrics Comparison. Similar results are achieved on INTENT-2 dataset.

4.2.2. I-rec Results

I-rec evaluates the coverage of query subtopics in the search results. The experiment results are shown in Figure 4. For both INTENT-1 and INTENT-2 datasets, *IA-Select(TopicLD)* achieves the best performance. *UserLD* is slightly worse than *IA-Select(TopicLD)*. *UserLD* is as well as *IA-Select(TermLD)* on INTENT-1 dataset, but *UserLD* is slightly better than *IA-Select(TermLD)* on INTENT-2 dataset.

UserLD and *IA-Select(TopicLD)* both try to maximize the coverage of query subtopics. The difference is that *UserLD* tries to maximize the coverage of user sessions in the hope that the covered user sessions reflect different query subtopics. However, *UserLD* avoids mining the query subtopics. *IA-Select* models subtopics explicitly and tries to cover as many subtopics as possible directly. *IA-Select* relies on a priori and good description of all query's subtopics. So it is reasonable that *IA-Select(TopicLD)* achieves the best performance in terms of *I-rec*. Nevertheless, the performance of *UserLD* proves the feasibility of promoting diversity on user session level.

We further analyze the possible reasons that *I-rec* of *UserLD* is worse than *IA-Select(TopicLD)*. First, some subtopics do not exist in the query logs. For example, the query "Transformers" (film series) contains 4 subtopics, i.e. "Transformers 1", "Transformers 2", "Transformers 3" and "Transformers 4". However, the query logs we used only contain "Transformers 1". Second, some queries and subtopics are unpopular which are seldom searched by users. For example, most users will never seek for information about the query "Polysilicon". *UserLD* relies on user search behaviors, so the loss or sparsity of query logs (which is usually not a problem for commercial search engines) may influence its performance. Third, the most possible reason is that the query logs we used are collected in only one month.

4.2.3. D#-metrics Results

We also evaluate *UserLD* with *D#-metrics*, which are more intuitive than other diversity metrics and promising for diversified IR evaluation according

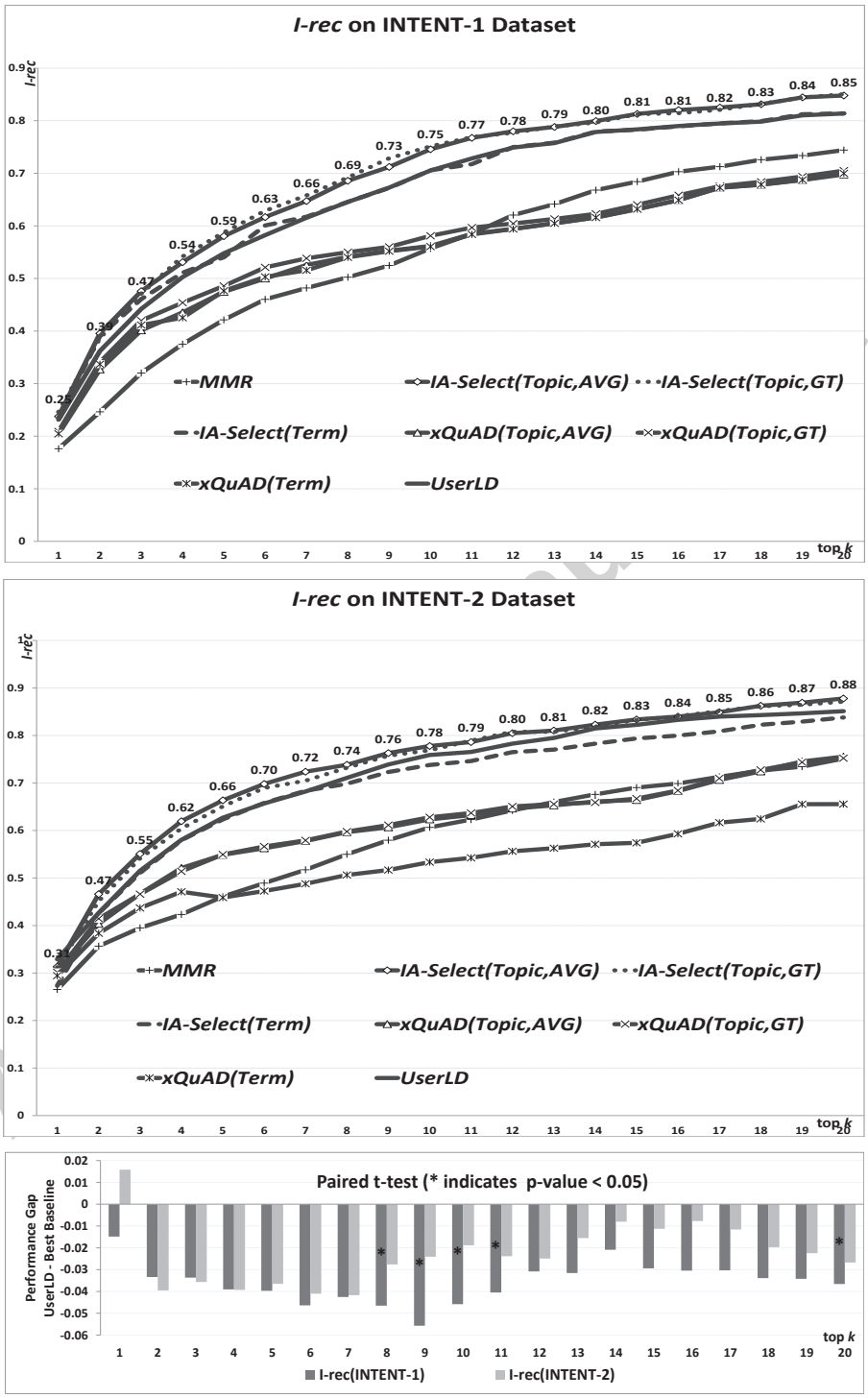


Figure 4: Result Diversity Comparison with *I-rec*.

to [42]. Different from *IA* metrics, *D-metrics* evaluate diversity by computing a global gain, i.e. $GG(r) = \sum_z P(z|q)g_z(r)$, for each document at rank r in search results. *D#-metrics* balances the effect of *I-rec* and *D-metric* with a parameter γ. As Sakai et al. [42] showed that the effect of the choice of γ on IR experiments is relatively small due to the fact that *I-rec* and *D-metrics* are already highly correlated with each other. Following their study, we also set $\gamma = .5$. The results are shown in Figure 5. As we can see, the results of *D#-metrics* are basically consistent with those of *IA* metrics which further confirms the improvement of *UserLD*.

4.3. Evaluation of Result Relevance

As diversity metrics measure per-subtopic documents relevance and favour documents covering many subtopics but not necessarily very relevant to the given query, so we further conduct experiments to analyze whether diversity is achieved at a cost of relevance. Four standard relevance-based metrics for web retrieval, *nDCG*, *nERR*, *Q-measure*, and *P@k* are used to evaluate the search results relevance. *nDCG*, *nERR* and *Q-measure* take into account the positions and relevance grades of search results in the top k list. *P@k* considers each ranking position as equally important and considers each document as either relevant or non-relevant. The evaluation results of the four metrics are shown in Figure 6. We can see that *UserLD* achieves the best performance. *IA-Select* and *xQuAD* are slightly less effective, and MMR gets the lowest performance.

The performance of *UserLD* significantly outperforms the baselines, especially on graded-relevance and position based metrics. The improvement comes from three aspects. First, the baselines consider document relevance to query subtopics merely, i.e., the similarity between the query subtopic z and the document d. *UserLD* models document relevance to user sessions, i.e., the query vector $TF\text{-}IDF(q)$ is expanded by considering user clicks $TF\text{-}IDF(d), d \in C(u)$. Second, *UserLD* models additional user behaviors, i.e., the weights $w(u)$ which take into account the query popularity $pop_{query}(u)$, the click popularity $pop_{click}(u)$ and the similarity $sim(u_c, u)$. Third, the baselines ignore relations

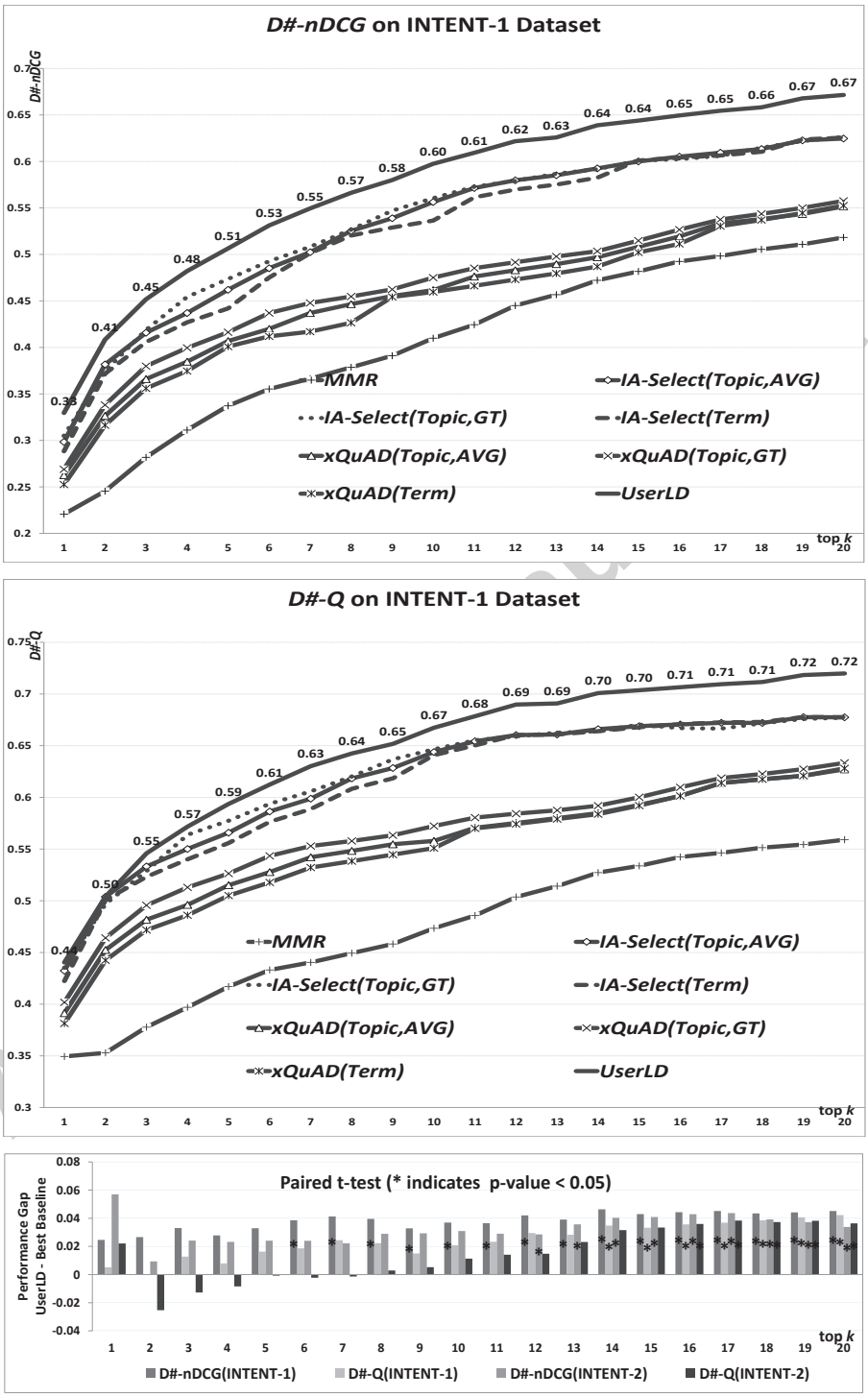


Figure 5: *D#-metrics* Comparison. Similar results are achieved on INTENT-2 dataset.

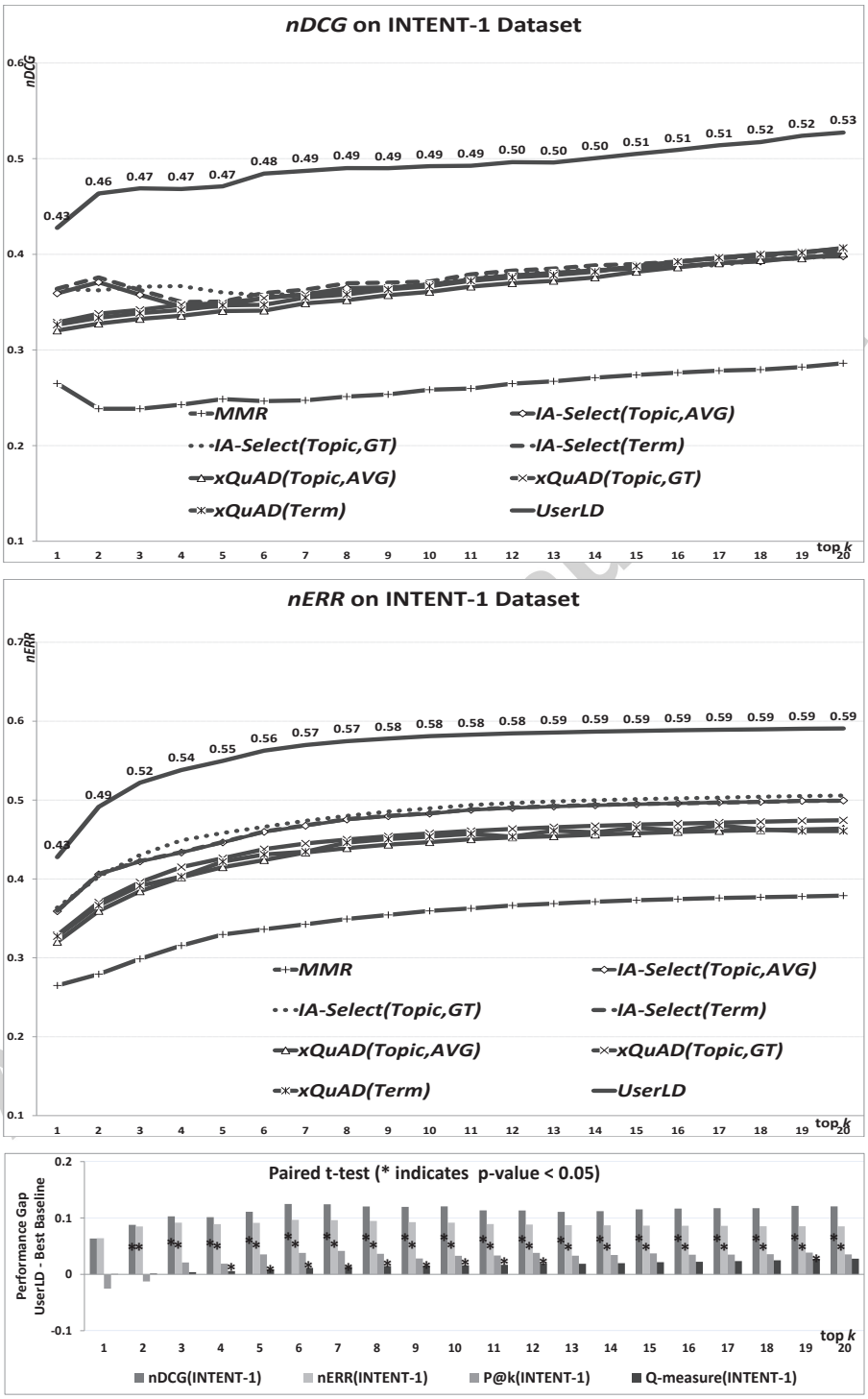


Figure 6: Result Relevance Comparison on INTENT-1 Dataset. Similar Q-*measure* and $P@k$ results are achieved. Similar results are achieved on INTENT-2 dataset.

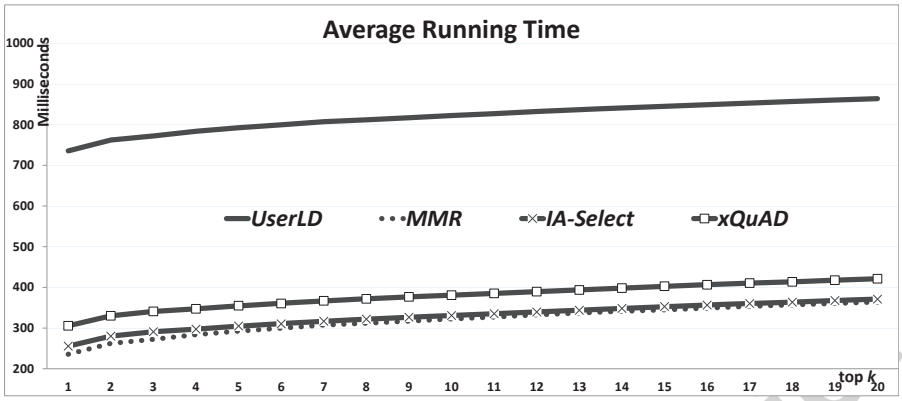


Figure 7: Practical Running Time on INTENT Dataset.

between user sessions and relations between documents. *UserLD* models this with $w(e_{(u\leftrightarrow u')})$ and $w(e_{(d\leftrightarrow d')})$.

4.4. Performance Summary

Overall, *UserLD* achieves the highest performance in terms of *IA* metrics and *D#-metrics*. As *IA* metrics and *D#-metrics* evaluate diversity by considering ranking relevance and subtopic coverage as a whole, so the improvement of *UserLD* lies in two aspects. First, the results relevance indicated by standard relevance-based metrics is improved greatly compared with the baselines. Second, the subtopic coverage indicated by *I-rec* is only slightly worse than a *TopicLD* model *IA-Select(TopicLD)* just because *IA-Select(TopicLD)* takes query subtopics in ground truth as input. The results mean that user search intents are indeed diversified and different user intents indeed reflect different subtopics of a query. In summary, *our approach has the ability of avoiding mining the query subtopics in advance while achieving almost the same or better performances compared with previous approaches.*

4.5. Practical Running Time

The practical running time of our approach on INTENT dataset is shown in Figure 7. The raw Greedy for our model is too slow to run out, so we did

not compare with it. As expected, the most time consuming task is done when choosing the first document. After that, the later documents can be chosen quickly. By comparison, as the number of search results k increases, raw Greedy slows down dramatically due to the extra computation stated in Session 3.3.

5. Conclusions and Future Work

In this paper, we propose *UserLD* and implement *UserLD* with a *Session Graph Construction* phase and a *Diversity Reranking* phase. Extensive experiments demonstrate the effectiveness of our approach, which confirms that *UserLD* can promote effective diversity while avoiding mining query subtopics or topic terms.

However, there are still at least two issues. First, only one-month query logs are available, so we do not know whether more query logs will further improve the performance or not. Second, although we adopt the Greedy algorithm, however the properties of the current diversity loss function cannot guarantee a $1 - \frac{1}{e}$ approximation. Either the diversity loss function or the optimization algorithm needs to be improved.

6. Acknowledgements

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), the Fundamental Research Funds of Shandong University, the Project Funded by the Priority Academic Program Development of Jiangsu Higer Education Institutions, and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology.

References

[1] H. Ma, M. R. Lyu, I. King, Diversifying query suggestion results., in: AAAI, AAAI Press, 2010.

[2] K. Raman, P. N. Bennett, K. Collins-Thompson, Toward whole-session relevance: exploring intrinsic diversity in web search, in: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM, 2013, pp. 463–472.

[3] D. Hong, L. Si, Search result diversification in resource selection for federated search, in: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM, 2013, pp. 613–622.

[4] K. Berberich, S. Bedathur, Temporal diversification of search results, in: Proceedings of the SIGIR 2013 workshop on time-aware information access, 2013.

[5] F. Sun, M. Wang, D. Wang, X. Wang, Optimizing social image search with multiple criteria: Relevance, diversity, and typicality, Neurocomputing 95 (2012) 40–47.

[6] P. Ren, Z. Chen, X. Song, B. Li, H. Yang, J. Ma, Understanding temporal intent of user query based on time-based query classification, in: Natural Language Processing and Chinese Computing, Springer, 2013, pp. 334–345.

[7] P. Ren, Z. Chen, J. Ma, Z. Zhang, L. Si, S. Wang, Detecting temporal patterns of user queries, Journal of the Association for Information Science and Technology.

[8] V. Dang, B. W. Croft, Term level search result diversification, in: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM, 2013, pp. 603–612.

[9] N. Naveed, T. Gottron, S. Staab, Feature sentiment diversification of user generated reviews: The freud approach, The freud approach.

[10] X. Liu, A. Bouchoucha, A. Sordoni, J.-Y. Nie, Compact aspect embedding for diversified query expansions., in: AAAI, 2014, pp. 115–121.

[11] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1998, pp. 335–336.

[12] C. X. Zhai, W. W. Cohen, J. Lafferty, Beyond independent relevance: methods and evaluation metrics for subtopic retrieval, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM, 2003, pp. 10–17.

[13] R. Agrawal, S. Gollapudi, A. Halverson, S. Ieong, Diversifying search results, in: Proceedings of the Second ACM International Conference on Web Search and Data Mining, ACM, 2009, pp. 5–14.

[14] R. L. Santos, C. Macdonald, I. Ounis, Exploiting query reformulations for web search result diversification, in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 881–890.

[15] Z. Dou, S. Hu, K. Chen, R. Song, J.-R. Wen, Multi-dimensional search result diversification, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 475–484.

[16] V. Dang, W. B. Croft, 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, 2012, pp. 65–74.

[17] J. Wang, J. Z. Huang, J. Guo, Y. Lan, Recommending high-utility search engine queries via a query-recommending model, Neurocomputing 167 (2015) 195–208.

[18] B. Huang, G. Yu, Research and application of public opinion retrieval based on user behavior modeling, Neurocomputing 167 (2015) 596–603.

[19] D. Beeferman, A. Berger, Agglomerative clustering of a search engine query log, in: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2000, pp. 407–416.

[20] X. Wang, C. Zhai, 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, 2007, pp. 87–94.

[21] Y. Hu, Y. Qian, H. Li, D. Jiang, J. Pei, Q. Zheng, Mining query subtopics from search log data, in: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, ACM, 2012, pp. 305–314.

[22] Y. Qian, T. Sakai, J. Ye, Q. Zheng, C. Li, Dynamic query intent mining from a search log stream, in: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, ACM, 2013, pp. 1205–1208.

[23] R. W. White, W. Chu, A. Hassan, X. He, Y. Song, H. Wang, Enhancing personalized search by mining and modeling task behavior, in: Proceedings of the 22nd international conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2013, pp. 1411–1420.

[24] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, G. Tolomei, Discovering tasks from search engine query logs, ACM Transactions on Information Systems (TOIS) 31 (3) (2013) 14.

[25] J. Wang, J. Zhu, Portfolio theory of information retrieval, in: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM, 2009, pp. 115–122.

[26] S. Liang, Z. Ren, M. De Rijke, Personalized search result diversification via structured learning, in: Proceedings of the 20th ACM SIGKDD inter-

national conference on Knowledge discovery and data mining, ACM, 2014, pp. 751–760.

[27] Y. Zhu, Y. Lan, J. Guo, X. Cheng, S. Niu, Learning for search result diversification, in: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, ACM, 2014, pp. 293–302.

[28] R. L. Santos, C. Macdonald, I. Ounis, Intent-aware search result diversification, in: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM, 2011, pp. 595–604.

[29] X. Chang, Y. Yang, E. Xing, Y. Yu, Complex event detection using semantic saliency and nearly-isotonic svm, in: Proceedings of the 32nd international conference on machine learning (ICML-15), 2015, pp. 1348–1357.

[30] X. Chang, Y. Yu, Y. Yang, E. Xing, They are not equally reliable: Semantic event search using differentiated concept classifiers, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[31] X. Chang, Y.-L. Yu, Y. Yang, A. G. Hauptmann, Searching persuasively: Joint event detection and evidence recounting with limited supervision, in: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, ACM, 2015, pp. 581–590.

[32] C. Khan, J. Lee, R. Blanco, Y. Chang, Predicting primary categories of business listings for local search ranking, Neurocomputing 168 (2015) 961–969.

[33] L. Nie, S. Yan, M. Wang, R. Hong, T.-S. Chua, Harvesting visual concepts for image search with complex queries, in: Proceedings of the 20th ACM international conference on Multimedia, ACM, 2012, pp. 59–68.

[34] R. Baeza-Yates, A. Tiberi, Extracting semantic relations from query logs, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2007, pp. 76–85.

[35] L. Nie, M. Wang, Z.-J. Zha, T.-S. Chua, Oracle in image search: A content-based approach to performance prediction, ACM Transactions on Information Systems (TOIS) 30 (2) (2012) 13.

[36] X. Song, L. Nie, L. Zhang, M. Liu, T.-S. Chua, Interest inference via structure-constrained multi-source multi-task learning, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2015, pp. 2371–2377.

[37] N. Craswell, M. Szummer, Random walks on the click graph, in: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2007, pp. 239–246.

[38] P. Ren, Z. Chen, J. Ma, S. Wang, Z. Zhang, Z. Ren, Mining and ranking users intents behind queries, Information Retrieval Journal 18 (6) (2015) 504–529.

[39] M. S. Charikar, Similarity estimation techniques from rounding algorithms, in: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, ACM, 2002, pp. 380–388.

[40] M. Minoux, Accelerated greedy algorithms for maximizing submodular set functions, in: Optimization Techniques, Springer, 1978, pp. 234–243.

[41] T. Sakai, The unreusability of diversified search test collections., in: EVIA@ NTCIR, Citeseer, 2013.

[42] T. Sakai, R. Song, 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, 2011, pp. 1043–1052.

Theorem 1. $L(R, G(u_c, q_c))$ is *non-negative, monotone* and *supermodular.*

Proof 1. *The* non-negative *property is obvious, since both* $w(u) \geq 0$ *and* $\overline{P}_{R \Rightarrow u} \geq 0$.

monotone*:* $\forall R^i \subseteq R^{i+1}$*, we have*

$$L(R^i) - L(R^{i+1}) = \sum_{u \in U_{q_c}} w(u)(\overline{P}_{R^i \Rightarrow u} - \overline{P}_{R^{i+1} \Rightarrow u})$$

$$= \sum_{u \in U_{q_c}} w(u) \left(\prod_{path \in path(R^i \Rightarrow u)} (1 - \prod_{e \in path} P(e)) \right)$$

$$\left(1 - \prod_{path \in path(R^{i+1} \setminus R^i \Rightarrow u)} (1 - \prod_{e \in path} P(e)) \right)$$

$$= \sum_{u \in U_{q_c}} w(u) \overline{P}_{R^i \Rightarrow u} (1 - \overline{P}_{R^{i+1} \setminus R^i \Rightarrow u}) \geq 0$$

supermodular*:* $\forall R^i \subseteq R^{i+1}$*, based on above deduction, we have*

$$L^1 = L(R^i) - L(R^i \cup \{d\}) = \sum_{u \in U_{q_c}} w(u) \overline{P}_{R^i \Rightarrow u} (1 - \overline{P}_{d \Rightarrow u})$$

$$L^2 = L(R^{i+1}) - L(R^{i+1} \cup \{d\})$$

$$= \sum_{u \in U_{q_c}} w(u) \overline{P}_{R^i \Rightarrow u} (1 - \overline{P}_{d \Rightarrow u}) \overline{P}_{R^{i+1} \setminus R^i \Rightarrow u}$$

As a result,

$$L^1 - L^2 = \sum_{u \in U_{q_c}} w(u) \overline{P}_{R^i \Rightarrow u} (1 - \overline{P}_{d \Rightarrow u})(1 - \overline{P}_{R^{i+1} \setminus R^i \Rightarrow u}) \geq 0$$