mNIR: Diversifying Search Results based on a Mixture of Novelty, Intention and Relevance

Reza Taghizadeh Hemayati, Laleh Jafarian Dehkordi, Weiyi Meng

Department of Computer Science, Binghamton University
Binghamton, NY 13902, USA
{hemayati, ljafari1, meng}@cs.binghamton.edu

ABSTRACT. Current search engines do not explicitly take different meanings and usages of user queries into consideration when they rank the search results. As a result, they tend to retrieve results that cover the most popular meanings or usages of the query. Consequently, users who want results that cover a rare meaning or usage of query or results that cover all different meanings/usages may have to go through a large number of results in order to find the desired ones. Another problem with current search engines is that they do not adequately take users' intention into consideration. In this paper, we introduce a novel result ranking algorithm (mNIR) that explicitly takes *result **n**ovelty*, *user **i**ntention-based distribution* and *result **r**elevancy* into consideration and mixes them to achieve better result ranking. We analyze how giving different emphasis to the above three aspects would impact the overall ranking of the results. Our approach builds on our previous method for identifying and ranking possible categories of any user query based on the meanings and usages of the terms and phrases within the query. These categories are also used to generate category queries for retrieving results matching different meanings/usages of the original user query. Our experimental results show that the proposed algorithm can outperform state-of-the-art diversification approaches.

Keywords: Search engine, result ranking, diversification, user intention.

1 INTRODUCTION

Search engines are widely used by users to find desired information. Unfortunately, modern search engines still do not satisfy many users' needs. Several reasons contribute to this problem. First, users tend to submit very short queries (most have only 1-3 terms), which can be ambiguous with different interpretations. Second, different users may have different search goals even when they submit the same query. Some search algorithms (e.g., PageRank [13]) tend to retrieve results that cover the most popular meanings/usages of query terms. For example, when "apple" was submitted to Google on May 24th, 2012, all search results in the first result page are related to the company Apple. This means that if a user wants to find results about some relatively rare meanings/usages of a query, the user probably has to go through a long list of results to find what he/she wants. For example, according to Wikipedia, the term "jaguar" may refer to a large cat, a car, a supercomputer, a type of military aircraft (SEPECAT Jaguar), etc. When this term is submitted as a query to Google on May 24th, 2012, the first result relevant to SEPECAT Jaguar was ranked at 333 (in the 33rd page with 10 results on each page).

One way to remedy the above problem is to intentionally select search results covering different possible meanings/usages of a query and show them among the top-ranked results. This is called *diversification*. In order to perform diversification well, we should address the following two issues. First, we need to identify different meanings/usages of a given query so we can find results for each meaning/usage and include them among the top results. Second, we need to know to what extent each of the meanings/usages match the *expected intention* of the user who submitted the query so we can diversify accordingly (e.g., include more results from the more likely intentions in the result lists).

In this paper, we propose a novel algorithm to select and rank search results (called mNIR) which tries to rank the search results by a ***m****ixture* of three result ranking preferences: *result* ***n****ovelty*, *user* ***i****ntention-based distribution* and *result* ***r****elevance*. *Result novelty* means that the ranking algorithm tries to include search results covering different meanings/usages among the top-ranked ones. This is also known as *result diversification*, which is beneficial for queries that have multiple meanings/usages. *User intention-based distribution* means that the ranking algorithm takes different possible intentions of the user into consideration when ranking the search results. When a query has multiple possible interpretations, it is often the case the likelihood for different interpretations (intentions) are different. *Result relevance* means that the ranking algorithm attempts to rank search results in descending order of their likelihood to be relevant to the query. In traditional information retrieval, relevance-based ranking is converted to similarity-based ranking in practice. The result ranking methods used in current search engines are mostly relevance-based ranking techniques.

In this paper, we assume that, for each query, its different possible meanings/usages and their likelihoods/probabilities have been estimated using an existing method [9].

This paper has the following contributions:

1. We introduce a novel search result diversification algorithm (called mNIR) by mixing three ranking preferences, namely, result novelty, user intention-based distribution and result relevance. By assigning different percentage values to these ranking preferences, our algorithm tries to produce the results that satisfy those ranking preference (based on the emphases given to them) as much as possible.
2. We perform extensive experiments to evaluate the effect of different mixing percentages on the performance of our proposed algorithm using a diverse set of measures. We also compare our algorithm with two existing state-of-the-art diversification algorithms. Our experimental results show that the new algorithm can achieve better overall performance.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 provides a brief review of the background needed for our approach. Section 4 presents the main steps of our approach. Section 5 reports our experimental results. Section 6 concludes the paper.

2 Related Work

Result diversification has received a lot of attention recently. In [3], the authors described the Maximal Marginal Relevance (MMR) problem, and they showed how a

trade-of between novelty and relevance of search results can be made explicit through the use of two functions, one measuring the similarity among documents and the other the similarity between documents and the query. Zhai et al. [22] stated that in general it is not sufficient to return a set of relevant results as the correlation among the returned results is also very important. They studied both novelty and relevancy in the language modeling framework. They developed an evaluation framework for subtopic retrieval based on the metrics of subtopic recall and subtopic precision. Agrawal et al. [1] proposed a diversification objective that tries to maximize the likelihood of finding a relevant document among the top-*k* results given the categorical information of the queries and documents. Furthermore, the authors generalized some classical IR metrics, including NDCG, MRR, and MAP, to evaluate their work for the value of diversification. They showed that their algorithm performs better in these generalized metrics compared to the results produced by commercial search engines.

Gollapudi et al. [8] defined a set of natural, intuitive axioms that a diversification system is expected to satisfy. The authors proved that all axioms cannot satisfy all these axioms simultaneously. Finally, they proposed an evaluation methodology to characterize the objectives and the underlying axioms. Rafiei et al. [12] modeled the diversity problem as expectation maximization and presented algorithms to estimate the optimization parameters. In [4], documents are selected sequentially according to relevance. The relevance is conditioned on documents having been already selected. Words in previous documents are associated with a negative weight to improve novelty. They considered different query intents, so that the result set covers multiple query intents. Clough et al. [7] examined user queries with respect to diversity. The authors found that a broad range of query types may benefit from diversification. They also showed that non-ambiguous queries can also need diverse search results. In [2], the authors studied the problem of diversifying search results by exploiting the knowledge derived from query logs. They presented a general framework for query result diversification using query recommendation algorithms, to detect ambiguous queries that would benefit from diversification, and to devise all the possible common specializations to be included in the diversified list of results along with their probability distribution. The different meanings and facets of queries are disclosed by analyzing user behaviors recorded in query logs. During the analysis, the popularity of the different specializations is also derived. The popularity distribution is then used to maximize the “usefulness" of the final set of documents returned.

Santos et al. [16] introduced the xQuAD probabilistic framework for search result diversification, which explicitly represented different query aspects as ‘sub-queries’. They defined a diversification objective based on the estimated relevance of documents to multiple sub-queries, as well as on the relative importance of each sub-query to the original query. Later they [17, 18] proposed a selective diversification approach. In particular, given an unseen query, their approach learns a trade-off between relevance and diversity, based on optimal trade-offs observed for similar training queries. As a result, their approach effectively determines when and how to diversify the results for an unseen query. Sakai et al. [14, 15] proposed an alternative way to evaluate diversified search results, given intent probabilities and per-intent graded relevance assessments.

Diversification has also been studied for purposes different from search engine result diversification. Radlinski et al. [11] directly learned a diverse ranking of results based on users' clicking behavior through online exploration. They maximize the probability that a relevant document is found among the top-*k* ranked results. Since users tend not to click similar documents, online learning produces a diverse set of documents naturally. This approach cannot readily be used for tail queries since it requires user feedback. Clarke et al. [6] studied diversification in question answering while Ziegler et al. [23] studied the problem from a "recommendation" point of view.

Our approach differs from the above methods in the following aspects. First, we leverage the previous works [9, 19] to find all possible intentions (meanings/usages) of a query based on Wikipedia and WordNet. This method also estimates the importance of each meaning/usage. Second, we introduce different ranking preferences (*result relevancy*, *result novelty*, and *user intention-based distribution*) that a diversification system should satisfy. We introduce an algorithm to satisfy these preferences in different ways. One of the objectives of this paper is to analyze the performance of our method by giving different emphases to each of the ranking preferences. Our algorithm follows a two-step process for selecting and ranking search result records (SRRs; each SRR consists of a URL, a title and a snippet) while taking the preferences into consideration. Finally, we propose a scheme to retrieve SRRs that can guarantee the retrieval of SRRs covering different meanings/usages of each query.

3 Background Review

In this section, we provide a brief review of the background needed by our approach. This review outlines how different meanings/usages of a query are obtained, how their probabilities are estimated and how the initial SRRs for each meaning/usage are retrieved.

1. *Alternative query generation* [9]. For each user query *Q*, this step generates a set of *alternative queries* (AQs). All AQs contain the same set of query terms that appear in *Q* but contain different phrases. For convenience of discussion, we will refer both query terms and phrases as ***concepts***.

2. *Definition category generation* [9]. A *definition category* (DC) [9] is a combination of *meanings* or *usages* derived from the concepts (query terms/phrases) of an AQ. This step generates all possible DCs for each AQ by combining one meanings/usage from each concept in the AQ.

***3. Definition category probability estimation* [9]**. This step estimates the probability of each DC (it represents a meaning/usage of query *Q*) generated in Step 2. The approach in [9] includes a method to compute the ranking score of any given DC (the score is denoted by *RS*(DC)). Let DC_1, …, DC_m be all the DCs generated in Step 2 and $total\_score = RS(DC_1) + \ldots + RS(DC_m)$. Then we define the probability of DC_k with respect to query Q to be $Pr(DC_k \mid Q) = RS(DC_k) / total\_score$.

4. *Definition category label generation* [19]. Given a DC as input, we use the method introduced in [19] to obtain a label for the DC. A label for a DC is a set of terms/phrases that summarizes the DC. This method uses Wikipedia and WordNet to

generate candidate labels. Wikipedia provides some useful information for each concept (like definitions, categories, disambiguation page, etc.), which can be used to generate candidate labels. WordNet provides information like synonyms, hypernyms, etc. that can also be used as candidate labels. These candidate labels are ranked for each DC and then the top-ranked candidate label is taken as the label for the DC.

5. *Input SRR generation*. For each DC, we use its label as a query, called *label query* or *category query*. We submit the query to a search engine (Google is used in this paper) to retrieve the top *n* SRRs (*n* is 10 in this paper). We denote this list of SRRs as SRR(DC). Note that the order of the SRRs in the list is important.

For each user query *Q*, the set of DCs, i.e. DC_1, …, DC_m, and the lists of SRRs retrieved by *Q* and the label queries of these DCs, i.e. SRR(Q), SRR(DC_1), …, SRR(DC_m), form the input SRRs to our result ranking algorithms. In our experiments, for comparison purpose, we also use the original user query to retrieve a list of top *k* SRRs using the same search engine.

To summarize, for the rest of this paper, we assume that for each user query, we have obtained its meanings and usages (i.e., the DCs), and for each such meaning/usage, we have obtained its probability (i.e., the likelihood this meaning/usage captures the intention of the user who submitted the query) and the list of SRRs (retrieved by the label query of the DC). Based on the above information, we develop our mNIR algorithm.

4 Selecting and Ranking SRRs

In this section, we present the algorithm mNIR that tries to mix the relative weights on the three ranking preferences when selecting and ranking search results. In Section 4.1, we define the three preferences more precisely. In Section 4.2, we describe the two-step process of the algorithm mNIR. In Section 4.3, we present our result selection method for the first step. In Section 4.4, we present our result ranking method for the second step.

4.1 Ranking Preferences

We define the ranking requirement for each of the three ranking preferences below. Let *Q* be the user query under consideration.

Result Novelty based Ranking. To satisfy this ranking preference, the displayed results should be as diverse as possible.

For example, suppose query Q has 5 meanings/usages. If 5 results are desired, then there should be one result for each of the 5 meanings/usages.

User Intention based Distribution Ranking. Given a probability distribution of query *Q* on different meanings/usages of *Q*, $\{Pr(DC_k \mid Q) \mid k = 1, \ldots, m\}$, to satisfy this ranking preference, the percentage of displayed results for DC_k among all displayed results should be equal to $Pr(DC_k \mid Q)$.

Novelty-based ranking and intention-based distribution ranking are closely related. The latter can be approximately considered as a weighted version of the former, i.e., in the latter approach, categories (DCs) that have higher probabilities will have more

related results selected for display while the former selects at least one result from each category (as much as possible). Another difference is that categories with very small probabilities are often ignored by the intention-based distribution ranking but less likely by the novelty-based ranking. Specifically, if $Pr(DC_i \mid Q) * n$ is significantly less than 1, where n is the number of desired results to be displayed, then DC_i will effectively be ignored by the intention-based ranking.

Result Relevance based Ranking. To satisfy this ranking preference, the results retrieved from a given corpus (e.g., the set of documents indexed by a search engine) need to be ranked in descending order of their likelihood to be relevant to Q. In this paper, the likelihood of a result's relevance to a query will be modeled by the similarity of the result with the query.

Popular similarity functions such as the *Okapi function* [13] and the *Cosine function* can be used to compute the similarity between a retrieved result and a query.

4.2 Two-Step Framework

In this paper, we assume that there is a way for users to specify their search preferences through the query interface of the search engine the user uses. In this paper, users can mix the three ranking preferences described in Section 4.1 by assigning different weights to them. The total weight of all ranking preferences should be equal to one all the time. To better support different preference mixes, we adopt a two-step framework for our algorithm. These two steps are described below.

- Step 1: *Result selection*. The goal of this step is to select a set of SRRs from the initial lists of SRRs, i.e., $SRR(DC_1)$, …, $SRR(DC_m)$, SRR(Q) (see Section 3).
- Step 2: *Result ranking*. Suppose n SRRs are to be displayed to the user. The goal of this step is to rank the n SRRs from the set of SRRs selected in Step 1, again, based on the ranking preference mix given by the user.

These two steps are introduced in the next two subsections.

4.3 Selection Method (SM)

We will select $m*n + \tau*n$ SRRs by sending the label queries generated from the categories (DC_1, …, DC_m) and the original query (Q) to a search engine. We retrieve n SRRs for each DC and $\tau*n$ SRRs for the original query (in this paper τ is 5). These SRRs will be used in the next step. The perfect selection will be the list that has the most relevant SRRs for each DC and Q. These SRRs will be ranked in the next step (Section 4.4). If only the original query Q is used to retrieve results, then to ensure results for some rare meanings/usages are obtained, we often need to retrieve hundreds or even thousands of SRRs. To overcome this problem we use the label query generated for each category [19] to retrieve relevant SRRs. But if only the label queries are used, we may miss some important relevant SRRs with respect to the original query. Thus, we use both label queries and the original query to collect initial results.

The ranking method of mNIR (see Section 4.4) will be based on a mixture of three ranking preferences. For each result that is retrieved by a category label query, we know its category (meaning/usage). This allows us to evaluate its novelty and whether it matches any given intention during the ranking process. In order to implement the

relevance-based preference, we need to know the relevance of each result. In practice, the likelihood of relevance of a result is estimated based on its similarity with the query (we use the original query *Q*). If a result SRR* retrieved by a category label query is not retrieved by *Q*, its similarity is unknown. By retrieving τ*n SRRs for the original query with τ > 1, we try to increase the chance that SRR* is retrieved by the original query *Q*. If SRR* is not included in the τ*n SRRs retrieved by *Q*, we assign a low relevancy score to it (the score is lower than the lowest score of the SRRs retrieved by *Q*).

4.4 Ranking Method (RM)

In this method, we assign a percentage value to each ranking preference. Let λ_1, λ_2, and λ_3 denote the percentage values assigned to the preferences of *result novelty*, *result relevancy*, and *user intention-based distribution*, respectively, and satisfy the following conditions:

$$\mathbf{0} \leq \lambda_1, \lambda_2, \lambda_3 \leq \mathbf{1} \text{ and } \lambda_1 + \lambda_2 + \lambda_3 = 1.$$

In this section we introduce a Ranking Method (RM) – a novel probabilistic ranking algorithm that explicitly mixes different preferences into the ranking process. This method is shown in Figure 1.

Given an ambiguous query *Q* and a set of SRRs *R* selected in the first step for this query, we build a new SRR ranking list *S* by iteratively selecting the *n* highest scored SRRs from *R* according to the following probability mixture model:

$$\lambda_1 \text{Nov}(\text{DC},Q) + \lambda_2 \text{Rel}(\text{d},Q) + \lambda_3 \text{Int}(\text{DC}|Q) \quad (1)$$

In particular, these probabilities can be regarded as modeling *result novelty*, *result relevancy*, and *user intention-based distribution*, respectively, with a mixing of parameters λ_1, λ_2 and λ_3 controlling the tradeoff among the three. In Expression (1), Int(DC|*Q*) is the likelihood of DC being the real intention behind user's query *Q* (See Steps 2&3 in Section 3), *Rel*(*d*, *Q*) is the likelihood of a document *d*'s relevance to *Q*, and Nov(DC, *Q*) is a binary value, it is one if there hasn't been any relevant SRRs selected for this DC, and it is zero once an SRR relevant to this DC is selected. Initially Nov(DC, *Q*) is one for all DCs. For the above model, when λ_1=1 and there is a tie among at least two SRRs, we use a secondary preference to break the tie. In this case we select an SRR from the higher weighted DC.

RM (n, Pr(DC|Q), λ_1, λ_2, R) // $\lambda_3 = 1 - (\lambda_1 + \lambda_2)$

1 *S← { }*
2 *Nov(DC, Q) ← 1*
3 *Int(DC| Q) ← Pr(DC| Q)*
4 *While |S| < n do*
5 *d*←arg max$_{d \in R \setminus S}$(λ_1Nov(DC, Q) + λ_2Rel(d, Q) + λ_3Int(DC| Q))*
6 *R← R – {d*}*
7 *add d* to the end of S*
8 *Nov(DC, Q) ← 0*
9 $\forall DC, if\ (Nov(DC, Q)) == 0)$
10 *$\lambda_2 \leftarrow ((\lambda_1/2) + \lambda_2)$; $\lambda_3 \leftarrow ((\lambda_1/2) + \lambda_3)$*

```
11              λ1 ← 0
12          Int(DC| Q) ← (n*Int(DC| Q)-1)/n
13      End while
14      Return S
```

Fig. 1. Ranking Method of mNIR

In each iteration once an SRR relevant to a DC from *R* is selected, we reduce Int(DC|Q) from that DC using Int(DC|Q)=(n*Int(DC|Q)-1)/n. Novelty for a DC changes from one to zero once an SRR has been selected from that DC. When the novelty score Nov(DC, Q) becomes zero, its percentage value (i.e., λ_1) is evenly distributed to λ_2 and λ_3.

5 Evaluation

In Section 5.1, we describe the dataset used in our evaluation study. We also introduce various performance measures. In Section 5.2, we review the algorithms to be evaluated and compared. In Section 5.3, we present the experimental results.

5.1 Dataset and Performance measures

The query dataset we use consists of 50 queries. These queries are from TREC 2010 Web track. Among the queries, the numbers of queries having 1, 2, 3 and 4 terms are 26, 11, 10 and 3, respectively. There are 2.59 categories per query on the average. We consider up to top 5 meanings/usages for each query. Some queries may have fewer meanings/usages.

Some traditional IR metrics like MAP and MRR are widely used to measure search quality. However, they do not take the quality of diversification into consideration. Several metrics, including the *k*-call metric [20] and a-NDCG [21], have been proposed for evaluating the diversity of search results. However, these metrics do not consider the relative importance of different categories (DCs) and how well a result matches a category.

To address this problem in this paper, we use Intent-Aware (IA) measures to evaluate the ranking algorithms with different ranking preferences. We use existing measures MAP-IA@n [1, 3], MRR-IA@n [1, 3] and NDCG-IA@n [3, 18] to evaluate different aspects of our ranking algorithms. We review these measures below.

MAP-IA@n (Intent Aware (IA) MAP) [1, 3]:

In order to define MAP-IA@n, we first define AP(DC)@n (Average Precision with respect to category DC) [1]. This measure is an adaptation of the AP measure in IR research [10]. This adaptation takes the results retrieved for different categories (DCs) into consideration, making the measure capable of also measuring the effectiveness of diversification. The average precision of a ranked result set for query *Q* and category DC_k for *n* search results is defined as:

$$AP(DC_k)@n = \frac{\sum_{j=1}^{n} P(j) * relevance(j)}{N(DC_k)} \qquad (2)$$

where j is the ranking position of a retrieved result, *relevance*(j) denotes the relevance of the j-th ranked result (it is 1 if the j-th ranked result is relevant with respect to DC_k, and 0 otherwise), $P(j) = \sum_{i=1}^{j} relevance(i)/j$, and $N(DC_k)$ denotes the number of relevant results for category DC_k among n search results.

Traditional IR metrics focus solely on the relevance of documents. Consider a query that belongs to two categories DC_1 and DC_2. Suppose $Pr(DC_2 \mid Q) >> Pr(DC_1 \mid Q)$, and we have two results, r_1 is rated excellent for DC_1 (but unrelated to DC_2), and r_2 is rated good for DC_2 (but unrelated to DC_1). As a result, the order (r_1, r_2), i.e., ranking r_1 ahead of r_2, will yield a higher AP score. Yet an "average" user will find the order (r_2, r_1) more useful. This is because AP treats all intentions (categories) equally. However, some categories are more likely to match users' expected intentions. In order to take into account the likelihood of user intention, we define intent aware MAP over all intentions of a query Q to be:

$$\text{MAP-IA}(Q)@\text{n} = \sum_{k=1}^{m} \Pr(DC_k \mid Q) * AP(DC_k)@n \quad (3)$$

The MAP-IA@n over all queries in the query set (SetQ) is defined as follows:

$$\text{MAP-IA@n} = \left(\sum_{Q \in SetQ} MAP - IA(Q)@n \right) \Big/ |SetQ| \quad (4)$$

where |SetQ| is the number of queries in SetQ.

NDCG-IA@n (Intent Aware (IA) NDCG) [1, 3]:

This measure was used in different works (e.g., [3, 18]) and it is an adaptation of the NDCG (normalized discounted cumulative gain [10]) for measuring the diversification of the search results produced by a search system. This adaptation takes the results retrieved for different categories (DCs) into consideration, making the measure capable of also measuring the effectiveness of diversification. $\text{NDCG}(DC_k)@n$ is defined as follows:

$$\text{NDCG}(DC_k)@\text{n} = \frac{\sum_{r=1}^{n} GG(r)/\log(r+1)}{\sum_{r=1}^{n} GG^*(r)/\log(r+1)} \quad (5)$$

where $GG(r)$ is the (cumulative) gain at rank r produced by an algorithm, $GG^*(r)$ is the (cumulative) gain at rank r in an ideal ranked list (the ideal list is the list built based on the intention distribution based on the probability of each category). In this paper, we consider two levels for gain value: 0 if the result is not relevant and 1 if it is relevant. In order to take into account the likelihood of user intention, we define intent aware NDCG of a given query Q using the following formula:

$$NDCG\text{-}IA(Q)@n = \sum_{k=1}^{m} P(DC_k \mid Q) * NDCG(DC_k)@n \quad (6)$$

The NDCG-IA@n over all queries in the query set SetQ is defined as follows:

$$NDCG - IA@n = \left(\sum_{Q \in SetQ} NDCG - IA(Q)@n \right) \Big/ |SetQ| \quad (7)$$

MRR-IA@n [1, 3]:

The reciprocal rank (RR) is the inverse of the position of the first relevant document in the ordered result list. This value is zero if there is no relevant document among retrieved documents. The mean reciprocal rank (MRR) of a query set is the average

reciprocal rank of all queries in the query set. Using the same idea of averaging user intention, we define intent aware MRR for a given query Q to be:

$$\text{MRR-IA}(Q)@\text{n} = \sum_{k=1}^{m} \Pr(DC_k \mid Q) * RR(DC_k)@n \tag{8}$$

where $\text{RR}(\text{DC}_\text{k})@\text{n}$ is reciprocal rank of the first result from DC_k. The MRR-IA@n over all queries in the query set SetQ is defined as follows:

$$MRR - IA@n = \left(\sum_{Q \in SetQ} MRR - IA(Q)@n \right) \Big/ \mid SetQ \mid \tag{9}$$

5.2 Algorithms to Be Evaluated

In this section, we list the algorithms we will evaluate and compare.

- mNIR: In Section 4 we introduced a two-step framework which diversifies SRRs based on three preferences, with a mixing of parameters λ_1, λ_2 and λ_3 controlling the tradeoff among the three.

We compare the above algorithm with the following two state-of–the-art algorithms:

- xQuAD [16]: Santos et al. [16] introduced the xQuAD probabilistic framework for search result diversification, which explicitly represented different query aspects as 'sub-queries'. They defined a diversification objective based on the estimated relevance of documents to multiple sub-queries, as well as on the relative importance of each sub-query to the original query. By introducing a tuning parameter, their approach has a capability to give different preferences to either relevance or novelty. To enable fair comparison (i.e., all algorithms are given the same set of initial results), in our implementation of xQuAD, we use the label queries generated from categories (DCs) as sub-queries (these label queries are also used in our algorithms to obtain the search results corresponding to different meanings/usages). The confidence of the relevance likelihood of a document to a sub-query is determined by the estimated relevance of the document to the category (DC). The probability likelihood of a sub-query to be the real intention of user's query is determined by the probability likelihood of a category (DC) to be the real intention of user's query.

  Here is the brief overview of the adapted algorithm. Given an ambiguous query Q and an initial ranking R produced for this query, a new ranking S will be built by iteratively selecting the n highest scored documents from R, according to the following probability mixture model:

$$(1-\lambda)P(d \mid Q) + \lambda \sum_{DC_k \in Q} \left[P(DC_k \mid Q) P(d \mid DC_k) \prod_{d_j \in S} (1 - P(d_j \mid DC_k)) \right] \tag{10}$$

  where $\text{P(d}|Q)$ is the relevance likelihood of a document to Q, and $\text{P}(\text{DC}_\text{k}|Q)$ is the likelihood of DC_k being the real intention of Q, $\text{P}(\text{d}|\text{DC}_\text{k})$ is the relevance likelihood of document d to DC_k. In particular, these two probabilities can be regarded as modeling relevance and diversity, respectively, with a parameter λ controlling the tradeoff between the two.

- IA-Select [1]: In [1], the results are retrieved using the original user query and are then divided into different categories using a classifier. Again, in order for the

comparison to be fair, in our implementation of IA-Select, the results for different categories are directly obtained using the label queries generated from categories (DCs). In [1], one result is selected to be ranked next at a time and, each time, the result with the highest product of its "relevance score" and a weight score of its category is selected. After a result from a category is selected, the weight of the category is reduced to increase the chance for a result from a different category to be selected next. The confidence of the classification of a document to a given class is determined by the estimated relevance of the document to the category (DC) that represents the class.

Algorithm mNIR differs from IA-Select and xQuad mainly on preferences considered. Approximately, IA-Select and xQuad consider only (expected) user intention (they call it novelty in their papers) and relevance while mNIR additionally considers real novelty. These two methods are more likely to miss less likely (rare) meanings/usages of queries than mNIR.

5.3 Experimental Results

In this section, we report the experimental results for the algorithms described in Section 5.2 using the performance measures introduced in Section 5.1. In particular, we aim to answer the following three questions:

1. Can we improve diversification performance by using our proposed algorithm?
2. Can adding intention-based distribution preference to novelty and relevancy preferences improve diversification performance?
3. How do different mixings of the three ranking preferences affect the performance of the algorithm mNIR?

We start our experiments by studying MRR-IA@n results to see which algorithm can diversify more results. Then later report MAP-IA@n and NDCG-IA@n results to see the diversification performance considering both relevancy and novelty for all algorithms. We start executing our algorithm with a full emphasis being given to the novelty preference and then eventually start to reduce the emphasis to the novelty and increase the emphasis to the other two preferences evenly. We repeat the above execution for relevancy preference and intention preference.

MRR-IA@n: The MRR-IA measure checks the position of the first result for each category. This means that this measure emphasizes the quality of diversification. The MRR-IA results for mNIR (different values of λ_1, λ_2 and λ_3 are tested), xQuAD (different values of λ are tested) and IA-Select with different *n*'s are shown in columns 5-7 of Table 1. mNIR performed the best among all algorithms when more emphasis is given to the novelty preference. This is due to the fact that it tries to select and rank SRRs as diversified as possible. When the same maximum emphasis is given to both novelty and intention preferences evenly (i.e., $\lambda_1=\lambda_3=0.5$ and $\lambda_2=0$), the algorithm performs as well as when $\lambda_1=1$. The reason is that we try to satisfy the distribution of different intentions of queries at each rank as much as possible. This produces more diversified results at higher ranks. xQuAD performs relatively better when more preference is given to novelty (larger values of λ place more emphasis on novelty). IA-Select performed relatively poorly compared with other algorithms.

Table 1. MRR-IA@n, MAP-IA@n, nDCG@n

*	n	r	i	**MRR**	**MRR**	**MRR**	**MAP**	**MAP**	**MAP**	**NDCG**	**NDCG**	**NDCG**
	λ1	λ2	λ3	**@3**	**@5**	**@10**	**@3**	**@5**	**@10**	**@3**	**@5**	**@10**
m	1	0	0	**0.654**	**0.688**	**0.688**	**0.512**	**0.486**	0.194	0.554	0.54	0.614
	0.8	0.1	0.1	0.618	0.654	0.654	0.464	0.444	0.178	0.538	0.536	0.602
	0.6	0.2	0.2	0.618	0.654	0.654	0.464	0.444	0.178	0.538	0.536	0.602
	0.4	0.3	0.3	0.492	0.514	0.514	0.464	0.444	0.178	0.464	0.466	0.49
	0.2	0.4	0.4	0.602	0.622	0.622	0.43	0.366	0.188	0.556	0.564	0.598
	0	0.5	0.5	0.582	0.588	0.602	0.394	0.338	0.178	0.526	0.532	0.596
	0	0.1	0.9	0.628	0.636	0.604	0.436	0.356	0.204	**0.588**	0.606	0.652
	0	1	0	0.462	0.472	0.5	0.33	0.31	0.206	0.452	0.474	0.538
	0.1	0.8	0.1	0.486	0.526	0.544	0.34	0.314	0.162	0.46	0.496	0.564
	0.2	0.6	0.2	0.548	0.564	0.584	0.37	0.334	0.172	0.496	0.514	0.584
	0.3	0.4	0.3	0.548	0.58	0.588	0.394	0.35	0.182	0.528	0.53	0.592
	0.4	0.2	0.4	0.62	0.656	0.656	0.466	0.402	0.176	0.54	0.542	0.602
	0.5	0	0.5	**0.654**	**0.688**	**0.688**	**0.512**	**0.446**	**0.196**	0.554	0.546	0.616
	0	0	1	0.622	0.628	0.598	0.438	0.36	**0.218**	**0.586**	**0.608**	**0.658**
	0.1	0.1	0.8	0.644	0.668	0.638	0.47	0.378	0.204	0.57	0.58	0.636
	0.2	0.2	0.6	0.65	0.67	0.664	0.51	0.368	0.202	0.548	0.578	0.622
	0.3	0.3	0.4	0.62	0.642	0.642	0.466	0.41	0.178	0.54	0.554	0.606
	0.4	0.4	0.2	0.584	0.616	0.608	0.464	0.392	0.174	0.54	0.548	0.6
	0.5	0.5	0	0.584	0.62	0.62	0.44	0.434	0.172	0.508	0.53	0.598
X	1	0	-	0.514	0.554	0.576	0.312	0.268	0.202	0.52	0.582	**0.65**
	0.8	0.2	-	0.53	0.57	0.588	0.354	0.326	0.198	**0.564**	**0.594**	0.634
	0.6	0.4	-	**0.59**	**0.596**	**0.616**	0.462	0.324	0.19	0.554	0.56	0.598
	0.5	0.5	-	0.554	0.56	0.58	0.382	0.284	0.17	0.52	0.514	0.576
	0.4	0.6	-	0.52	0.556	0.572	**0.416**	**0.324**	0.168	0.482	0.5	0.572
	0.2	0.8	-	0.462	0.472	0.5	0.33	0.31	0.162	0.452	0.474	0.546
	0	1	-	0.462	0.472	0.5	0.33	0.31	**0.206**	0.452	0.474	0.538
I	-	-	-	**0.514**	**0.554**	**0.482**	**0.312**	**0.268**	**0.202**	**0.52**	**0.582**	**0.408**

** m=mNIR, X=XQUAD, I=IA-Select, n=Novelty, r=relevancy, i=intention*

MAP-IA@n: The MAP-IA results for mNIR, IA-Select and xQuAD with different *n*'s are shown in columns 8-10 of Table 1. It can be seen that mNIR performed the best among all algorithms when more emphasis is given to both novelty and intention preferences evenly ($\lambda_1=\lambda_3=0.5$). When more emphasis is given to the novelty preference ($\lambda_1=1$) mNIR also performs very well for MAP-IA@n when n=3 and n=5. mNIR performs the best when $\lambda_3=1$ for MAP-IA@10. We have observed that algorithms generally perform relatively well if they can cover most categories first by showing one result from each category. Once they cover most categories, they will perform better if they rank SRRs from more likely intentions first. SRRs from more likely intentions have a better chance to be relevant to the query. This can explain the behavior we observe in this experiment. This is due to the fact that by giving more preference to intention-based distribution, we are able to show more SRRs from more

likely intentions and rank these SRRs higher. This will give us a higher score for NDCG-IA and MAP-IA. The algorithms that consider only the *novelty* and *relevancy* preferences but not have a separate intention-based distribution preference (such as xQuad and IA-Select) do not have this benefit.

NDCG-IA@n: The results for NDCG-IA@n for all algorithms are shown in columns 11-13 of Table 1. In this evaluation, the ideal ranking list is the list built based on the expected intentions of each query. It can be seen that mNIR has the overall best performance for this measure when more emphasis is given to the intention-based distribution preference. xQuad ($\lambda = 0.8$ and $\lambda = 1$) also performed very well overall.

In this section, we investigated the performance of our proposed algorithm along with two state-of-the-art approaches (i.e., xQuAD and IA-Select). We used three major measures (MRR-IA@n, NDCG-IA@n and MAP-IA@n) to compare the performance of these algorithms. Our experiments show that our proposed algorithm can improve diversification performance without sacrificing the relevancy. By defining three parameters to assign weight to our ranking preferences, our proposed algorithm has the capability and flexibility to perform under different objectives. When the goal is to diversify the results as much as possible our algorithm performs the best when more emphasis is given to the novelty preference among all algorithms studied in this paper. When the goal is to have a balance between novelty and relevancy our algorithm still performs the best when more emphasis is given to intention-based distribution preference. We have shown that by adding user intention based preference to our algorithm, we are able to improve the diversification performance.

6 Conclusion

In this paper, we investigated the problem of diversifying search results based on three different ranking preferences (result novelty, user intention and result relevance) and proposed a novel algorithm (called mNIR) to select and rank search results which tries to rank the search results by a mixture of three result ranking preferences. By assigning different percentage values to these ranking preferences, our algorithm has the flexibility to produce the results that satisfy those ranking preference (based on the emphases given to them) as much as possible. We evaluated the proposed algorithm by assigning different percentage values to each of the ranking preferences together with two state-of-the-art algorithms (xQuAD and IA-Select) using a variety of performance measures. Our experimental results show that the proposed algorithm in this paper performs better than both existing algorithms. In the future, we plan to study the performance of our algorithm for different type of queries. In particular, given a diversification approach and an unseen query, we would like to develop a method to predict an effective mixture of the ranking preferences based on previously seen similar queries.

7 References

1. R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. ACM Intl Conf on Web Search and Data Mining, 2009.

2. G. Capannini, F.M. Nardini, R. Perego, F. Silvestri. Efficient diversification of web search results. PVLDB, 4(7), April 2011.(14)
3. J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. ACM SIGIR, pp.335-336, 1998.
4. O. Chapelle, S. Ji, C. Liao, E. Velipasaoglu, L. Lai and S. Wu. Intent-based diversification of web search results: metrics and algorithms, Information Retrieval Journal, 2011.
5. H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. ACM SIGIR, pp.429-436, 2006.
6. C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Buttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. ACM SIGIR, pp.659-666. 2008.
7. P. Clough, M. Sanderson, M. Abouammoh, S. Navarro, and M. Paramita. Multiple approaches to analysing query diversity. ACM SIGIR, pp.734-735, 2009.
8. S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. WWW Conference, pp.381-390, 2009.
9. R. T. Hemayati, W. Meng and C. Yu. Identifying and Ranking Possible Semantic and Common Usage Categories of Search Engine Queries. International Conference on Web Information System Engineering (WISE), 2010.
10. K. Järvelin and J. Kekäläinen: Discounted Cumulated Gain. Encyclopedia of Database Systems 2009: 849-853.
11. F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. ICML, pp.784-791, 2008.
12. D. Rafiei, K. Bharat, and A. Shukla. Diversifying web search results. WWW Conference, pp.781-790, 2010.
13. S. Robertson, S. Walker, M. Beaulieu. Okapi at Trec-7: Automatic Ad Hoc, Filtering, Vlc, and Interactive Track. 7th Text REtrieval Conference, 1999, pp.253-264.
14. T. Sakai, N. Craswell, R. Song, S. Robertson, Z. Dou, and C.-Y. Lin.Simple evaluation metrics for diversified search results. EVIA 2010, pp.42–50, 2010.
15. T. Sakai, R. Song. Evaluating Diversified Search Results Using Per-intent Graded Relevance. ACM SIGIR, 2011.
16. R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for Web search result diversification. WWW Conference, pp.881–890, 2010.
17. R. L. T. Santos, C. Macdonald, and I. Ounis. Selectively diversifying Web search results. ACM CIKM, pp.1179–1188, 2010.
18. R. L. T. Santos, C. Macdonald, and I. Ounis. Intent-aware search result diversification. ACM SIGIR, 2011.
19. R. T. Hemayati, W. Meng, C. Yu. Categorizing Search Results Using WordNet and Wikipedia. International Conference on Web-Age Information Management (WAIM), 2012
20. Y. Xu and H. Yin. Novelty and topicality in interactive information retrieval. J. Am. Soc. Inf. Sci. Technol., 59(2):201-215, 2008.
21. C. Zhai. Risk Minimization and Language Modeling in Information Retrieval. PhD thesis, Carnegie Mellon University, 2002.
22. C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. ACM SIGIR, pp.699-708, 2003.
23. C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. WWW Conference, pp.22-32, 2005