Beyond Greedy Search: Pruned Exhaustive Search for Diversified Result Ranking

Yingying Wu
Department of Mathematics
The University of Texas at Austin
ywu@math.utexas.edu

Yiqun Liu
DCST, Tsinghua University
Beijing, China
yiqunliu@tsinghua.edu.cn

Fei Chen
DCST, Tsinghua University
Beijing, China
chenfei27@gmail.com

Min Zhang
DCST, Tsinghua University
Beijing, China
z-m@tsinghua.edu.cn

Shaoping Ma
DCST, Tsinghua University
Beijing, China
msp@mail.tsinghua.edu.cn

ABSTRACT

As a search query can correspond to multiple intents, search result diversification aims at returning a single result list that could satisfy as many users' information needs as possible. However, determining the optimal ranking list is NP-hard. Several algorithms have been proposed to obtain a local optimal ranking with greedy approximations. In this paper, we propose a pruned exhaustive method to generate better solutions than the greedy search. Our approach is based on the observations that there are fewer than ten subtopics for most queries, most relevant results cover only a few subtopics, and most search users only focus on the top results. The proposed pruned exhaustive search algorithm based on ordered pairs (PesOP) finds the optimal solution efficiently. Experimental results based on TREC Diversity and NTCIR Intent task datasets show that PesOP outperforms greedy strategies with better diversification performance. Compared with the original non-pruned exhaustive search, the PesOP algorithm decreases the computational cost while maintaining optimality.

KEYWORDS

Retrieval models; Search process; Web search

ACM Reference Format:

Yingying Wu, Yiqun Liu, Fei Chen, Min Zhang, and Shaoping Ma. 2018. Beyond Greedy Search: Pruned Exhaustive Search for Diversified Result Ranking. In *ICTIR '18: 2018 ACM SIGIR International Conference on the Theory of Information Retrieval, September 14–17, 2018, Tianjin, China,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3234944.3234967

1 INTRODUCTION

Web search users expect to find relevant information within the top-ranked results [25]. However, when submitting one query, users may have multiple search intents [1, 23, 26], and it is thus difficult for the search engine to return a result list that covers all user intents (ambiguous queries) or aspects (underspecified or broad queries [30]) in the top positions. Given an ambiguous or underspecified query, search result diversification aims to produce a search engine result page (SERP) that maximizes the probability of satisfying different users' information needs [9, 11, 16, 25, 26, 28]. However, search result diversification has been proven to be NP-hard [4]. Several greedy search algorithms such as IA-Select [1] and xQuAD [27] are therefore proposed to find an approximation of the optimal diversified ranking list.

Table 1: An example of search result diversification in which the greedy strategy fails to produce optimal results.

Document	Subtopic 1 (importance = 0.5)	Subtopic 2 (importance = 0.5)
a	0.6	0.6
b	1.0	0
c	0	1.0

For example, consider three documents relevant to two subtopics weighted equally for some query, as shown in Table 1. If we choose the commonly-used weighted sum of document gains for subtopics as evaluation metrics [1, 8] (see Section 3 for details), the greedy search fails to produce optimal results. For example, to return a result list with length two, the algorithm will either return $\{a, b\}$ or $\{a, c\}$ because the diversified gain of a is 0.6, which is larger than that of b or c, which are both 0.5. However, with an exhaustive search strategy, we find that either $\{b, c\}$ or $\{c, b\}$ generate the largest possible weighted sum of document gains with α-NDCG [8] as the evaluation metric, as shown in Section 4. This example shows that the greedy strategy cannot always produce an optimal result list. However, the exhaustive search is impractical for online Web search scenarios due to its time complexity.

Mei [16] proposed DivRank, which is an explainable ranking algorithm based on a reinforced random walk. The model balances diversity and prestige to provide non-redundant and high coverage information in the top-ranked results. In the context of top-k recommendations, there have been advances in the improvement of greedy algorithms to diversify results for better recommendation results [15, 24]. Wang et al. [32] and Zuccon et al. [40] studied local search strategies to find top-k search result combinations to better

ICTIR '18, September 14–17, 2018, Tianjin, China
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5656-5/18/09…$15.00
https://doi.org/10.1145/3234944.3234967

satisfy users' information needs. However, these top-k recommendation strategies cannot be directly utilized to solve the diversified search problem because most of them diversify result combinations by taking the inter-result correlation into consideration. In top-k recommendation search problems, it is crucial to include novel results that are different from each other. In contrast, in diversified search formulation, result lists should be diversified to meet the information needs contained in different subtopics, thus generating result lists which best cover subtopics. Therefore, these strategies do not apply to search diversification. Hence, most research on diversified search such as TREC diversity and NTCIR Intent tasks still turns to greedy search strategies to generate diversified ranking lists. A recent FnTIR survey on search result diversification provides a complete discussion on this topic [29].

To improve the performance of result diversification, we propose a search algorithm that produces better lists than the greedy search with improved computational efficiency compared to exhaustive search. The diversification problem is NP-hard [4]; fortunately, a large proportion of documents are relevant to only one subtopic. Take the TREC Web Diversity and NTCIR Intent datasets as an example; only about 27% of retrieved documents are relevant to more than one subtopic, as discussed in Section 4. Therefore, a majority of candidate documents can be easily put into several sets of ordered pairs (see Definition 4.1) reflecting their relative orders in the optimal diversified result list. Rather than performing an exhaustive search on all ordered pairs in a set, effective pruning can be used to remove all the branches contradicting the determined ordered pairs. Following this idea, we propose a pruned exhaustive search algorithm based on ordered pairs (PesOP). The main contributions of this paper are as follows:

(1) We show that a large proportion of documents are relevant to only one subtopic, and prove that narrow subtopic coverage leads to more ordered pairs, which offers an opportunity to improve the efficiency of the exhaustive search in practice.
(2) We propose a pruned exhaustive search algorithm that exploits the above observation, leading to optimal diversified ranking with improved performance.
(3) Experiments on TREC and NTCIR collections show that the new search strategy produces better rankings than the greedy search, and the new strategy has improved computational efficiency compared to exhaustive search.

The rest of this paper is organized as follows: Section 2 reviews related work on diversification algorithms, Section 3 shows preliminary aspects of this study, Section 4 provides observations based on TREC and NTCIR datasets, Section 5 presents the PesOP algorithm, Section 6 reports experimental results and provides corresponding analysis by comparing PesOP with different search strategies, and Section 7 presents conclusions and directions for future work.

2 RELATED WORK

Given an ambiguous or underspecified query, search result diversification aims to produce a SERP that maximizes the probability of satisfying a general user's information needs. Existing strategies for search result diversification can be classified based on whether the adopted aspect representation is explicit or implicit, where the former has an explicit subtopic list provided.

2.1 Implicit Diversification Strategy

Implicit diversification does not require a pre-defined subtopic list [27]. Novelty-based diversification methods [34, 35] select the document that introduces the most novelty. A typical example is the maximal marginal relevance (MMR) method [3], which iteratively selects one document that is most relevant to the query and least similar to the documents already selected. Wang et al. [32] proposed a diversified search framework based on the portfolio theory that ranks results according to both relevance and variance. Some researchers use topic models to partition the candidate documents into clusters [14], and the importance of clusters is then adopted in the diversified ranking process with an MMR-like strategy.

To improve the efficiency of comparison among different documents in the non-diversified ranking list, several methods are proposed. For instance, a novelty-based diversification is modeled as a similarity search problem in a metric space [13]. In the study, three different kinds of strategies (pivoting-based, clustering-based, and permutation-based) are compared on the TREC Web track data. Implicit aspects are extracted with relevance modeling and topic models, then documents in each aspect are selected and added to the final ranking list [5].

2.2 Explicit Diversification Strategy

Explicit diversification relies on a list of subtopics corresponding to the possible search intents or aspects for a query. Given a list of possible subtopics (or sub-intents, which are weighted according to their popularity or importance) for a particular ambiguous or broad query, search result diversification can be cast as a maximum coverage problem [1] and hence is NP-hard. Therefore, most existing research efforts use a greedy algorithm to obtain an approximation.

While an approach can exploit a manually constructed subtopic list, several studies [27, 34, 35] have tried to identify the subtopics automatically. The NTCIR Intent and IMine tasks explicitly address this problem [26, 31]. Based on the generated subtopic list, the IA-Select [1], PM-2 [10], and xQuAD [27] algorithms select documents at each iteration with the highest diversified gain value, which is a weighted sum of gain for each subtopic. Although the principle is closely related to the novelty search algorithms [38, 39], an important difference is that the novelty search algorithms try to avoid redundancy among the selected documents, whereas the IA-Select, PM-2, and xQuAD algorithms aim to maximize the coverage of users' information needs [12]. Furthermore, Wu et al. [33] found that the subtopic distribution of result lists retrieved by explicit diversification algorithms deviates from the actual user intention distribution due to the objective function of the diversification problems.

The election-based diversification approach [10] considers the popularity of subtopics underlying a query. In each iteration step, it determines which subtopic is most important to be covered and then selects the document that is most relevant to this particular subtopic. Because a result may be relevant to more than one subtopic, this algorithm combines the relevance to the most important subtopic and the relevance to other subtopics. The above studies usually focus on developing new selection criteria, while the selection process relies on greedy search, which tries to iteratively select one document that presents the highest gain according to the selection function. Although there are some recent works in which researchers tried to replace the greedy search strategy with other solutions such as rank aggregation [19], most existing works rely on the greedy approach and focus their efforts on how to better estimate the diversified gain produced by documents. For example, Capannini [2] uses the similarity between documents in non-diversified ranking and documents in ranking lists generated by subtopics to re-rank the original non-diversified list. The re-ranking process, however, is also a greedy process. Both implicit diversification and explicit diversification require a better solution than the greedy search. As the exhaustive search is known to be intractable, our goal is to propose algorithms that can produce better results than the greedy search with reasonable time complexity. Our solution is to prune unnecessary search branches to reduce the computation time.

2.3 Pruned Exhaustive Search

Exhaustive search is the problem-solving methodology that enumerates all possible candidates for the solution and checks whether each candidate satisfies the problem's statement. Although the exhaustive search is simple to implement, and will always find a solution if one exists, the number of candidates is prohibitively large for real-world problems. Therefore, some strategies are proposed to prune the search space by reducing the number of candidate solutions without compromising performance. For example, the alpha-beta pruning algorithm [21] is widely used in machine playing of two-player games and seeks to decrease the number of nodes in the search tree. Other algorithms are also proposed to replace alpha-beta pruning with better efficiency without sacrificing accuracy, such as SCOUT [20] and MTD-f [22]. Recently, Neumann proposed a pruning method for text localization and recognition in real-world images based on character sequence information [18]. Chapelle et al. [6] studied the intent-aware search result diversification problem pruning branches according to the upper bound and lower bound for some set of candidates. Yu et al. [36] developed efficient diversification algorithms with a similarity threshold-based pruning strategy. From a more theoretical approach, Yuan [37] studied the diversified top-k clique search problem, where unpromising partial cliques are pruned to reduce the computational cost. There is also a survey on searching and pruning by Morrison [17]. In this paper, we focus on the explicit diversification strategy with a pre-defined list of subtopics for each ambiguous query topic.

3 PRELIMINARIES

Given a query q and its set of subtopics $C = \{s_1, \ldots, s_M\}$, we can generate M ordered document lists with a retrieval system as the initial list, where each list D_i contains documents ranked in decreasing relevance in the corresponding subtopic s_i for $1 \leq i \leq M$, and $D = D_1 \cup \ldots \cup D_M$ stands for the set of all candidate result documents for a query. When a diversified result list S is considered to be optimal according to a certain evaluation metric, it means that S receives the highest score among all possible lists. Therefore, it is necessary to define a reasonable evaluation metric that is used to estimate the diversified gain generated from candidate documents.

Algorithm 1: Greedy strategy for diversified search

```
Input: All retrieved documents $D$, the required list length $L$.
$S \leftarrow \emptyset$
while $|S| < L$ do
    $d' \leftarrow \mathrm{argmax}_{d \in D \setminus S}\, G(d, |S| + 1, S)$
    $D \leftarrow D \setminus \{d'\}$
    $S \leftarrow S \cup \{d'\}$
end
return $S$
```

A number of diversified evaluation metrics (e.g., NDCG-IA and MAP-IA [1]) estimate the diversified gain of a candidate document according to both its relevance to different subtopics and its position in the ranking list. Some other existing metrics further take into account the documents ranked higher than the current candidate (e.g., α-NDCG [8], ERR-IA [7], $D\#$-NDCG [25] and $DIN\#$-NDCG [25]). In general, the diversified gain of a document could be considered as a function of the document's relevance, the document's rank, and the influences from higher-ranked documents. Thus, the diversified gain for a certain document d in most diversified search evaluation metrics can be formulated as:

$$G(d, r, S') = \sum_{s_i \in C} (w_i \cdot g_i(d) \cdot decay_i(S') \cdot decay(r)), \tag{1}$$

where C is the subtopic list of a query, w_i is the weight attributed to a subtopic s_i, $g_i(d)$ is the gain value of the current document d for intent s_i, r is the current ranking position, S' is the set of documents ranked before d, $g_i(d)$ is the annotated relevance score of d to s_i, $decay_i(S')$ is the decay factor derived from S' for subtopic s_i, and $decay(r)$ stands for the decays with respect to a result's ranking position r. In most existing metrics (e.g. α-NDCG), $decay_i(S')$ is defined as the number of documents in S' that are relevant to s_i. We denote the documents ranked higher than the l-th document d_l as S'_l, and then the score of list S can be computed by accumulating the diversified gains of all d in S as

$$Score(S) = \sum_{l=1}^{|S|} G(d_l, l, S'_l).$$

Under different user behavior assumptions, $G(d, r, S')$ can be instantiated in different forms, and this will lead to different diversified ranking algorithms or evaluation metrics. For example, the function $G(d, r, S')$ in IA-Select is defined as:

$$G(d, r, S') = \sum_{s_i \in C} P(s_i|q) V(d|q, s_i) \prod_{d_j \in S'} (1 - V(d|q, s_i)),$$

where the factors $\prod_{d_j \in S'}(1 - V(d|q, s_i))$, $P(s_i|q)$, and $V(d|q, s_i)$ correspond respectively to $decay_i(S')$, w_i, and $g_i(d)$ in Equation (1). Although IA-Select iteratively selects the document with the highest score, it does not take into account the ranking position of the documents, thus the factor $decay(r)$ is dropped. $G(d, r, S')$ can be instantiated with α-NDCG as follows:

$$G(d, r, S') = \sum_{s_i \in C} J(d, s_i)(1 - \alpha)^{n_{i,S'}} / \log_2(r + 1), \tag{2}$$

where $J(d, s_i)$ indicates whether d is relevant to a particular subtopic s_i (or a nugget as in [8]) , the parameter $n_{i,S'}$ stands for the number of documents in S' that are relevant to s_i, and α represents the probability of an annotation error for documents annotated to be relevant. $J(d, s_i)$, $1/\log_2(r+1)$, and $(1-\alpha)^{n_{i,S'}}$ correspond respectively to $g_i(d)$, $decay(r)$, and $decay_i(S')$ in Equation (1). The weight of all subtopics in α-NDCG is set to be 1.

The gain value defined in Equation (2) assumes that subtopics underlying a query are distributed uniformly and a document has a binary relevance score to a subtopic. This definition works well when binary annotation is used as in some early TREC tasks, but it does not work with graded annotations or relevance scores annotated in $\mathbb{R}$. Therefore, we revise α-NDCG as:

$$G(d,r,S') = \sum_{s_i \in C} P(s_i|q)P(d|q,s_i)(1-\alpha)^{n_{i,S'}}/\log_2(r+1). \qquad (3)$$

In Equation (3), $J(d, s_i)$ is replaced with $P(d|q, s_i)$, and the importance of a subtopic w_i is replaced with $P(s_i|q)$. We notice that the revised definition also follows the evaluation framework in the original α-NDCG paper [8] which tries to estimate document relevance and subtopic importance as well. In the paper, $J(d, s_i)$ and equal importance scores are adopted instead of $P(d|q, s_i)$ and $P(s_i|q)$ to account for the typical TREC-like annotation standards at that time. However, diversity search benchmarks from NTCIR Intent and IMine tasks show that it is possible to estimate multi-grade relevance and subtopic importance scores with manual efforts. Another example of the gain definition of $G(d, r, S')$ is the one adopted by ERR-IA. According to Equation (1), its gain can be formulated as:

$$G(d,r,S') = \sum_{s_i \in C} P(s_i|q)R_i(d)\prod_{d_j \in S'}(1-R_j(d))/r, \qquad (4)$$

where $R_i(d)$ is the relevance of document d with respect to subtopic s_i. The factors $\prod_{d_j \in S'}(1-R_j(d))$, $P(s_i|q)$, $R_i(d)$, and $1/r$ correspond to $decay_i(S')$, w_i, $g_i(d)$, and $decay(r)$ in Equation (1), respectively. The revised version of α-NDCG as in Equation (3) will be adopted for the rest of the paper.

4 ORDERED PAIRS IN DIVERSIFIED SEARCH

As in the earlier example, greedy algorithms often fail to find the optimal solution. This occurs especially in cases where two or more documents have contradictory orders in different subtopics. For example, in Table 1, result b is more relevant than result a to Subtopic 1, but the reverse is true for Subtopic 2. If we use Equation (3) to evaluate the diversified gain, the output of Algorithm 1 will be $\{a, b\}$ or $\{a, c\}$ because $G(b, 1, \emptyset) = G(c, 1, \emptyset) = 0.5$ and $G(a, 1, \emptyset) = 0.6$, so a will be selected first. However, the optimal result list should be $\{b, c\}$ or $\{c, b\}$ because $Score(\{b, c\}) = Score(\{c, b\}) = 0.816$ while $Score(\{a, b\}) = Score(\{a, c\}) = 0.726$ according to Equation (3).

Suppose that we have two documents d_1 and d_2 that are relevant to $C'(C' \subset C)$ where C is the set of subtopics. For each subtopic $s_i \in C'$, the corresponding relevance scores are $rel_{k,i}$ $(k = 1, 2; 0 < rel_{k,i} \leq 1)$. If for some $s_i \in C_1$, we have $rel_{1,i} > rel_{2,i}$; while for some other $s_j \in C_2$, we have $rel_{1,j} < rel_{2,j}$ $(C_1 \neq \emptyset, C_2 \neq \emptyset, C_1 \cap C_2 \neq \emptyset)$, it will be difficult to determine which document should be placed higher in the final ranking list, since it depends collaterally on documents currently selected as well as later ones to be placed. However, the notion of ordered pairs improves the search efficiency in an exhaustive search.

Table 2: The distribution of documents retrieved for different numbers of subtopics in NTCIR-9/10 Intent tasks and TREC 2012 Web track diversity datasets.

#(subtopic retrieved)	NTCIR-9	NTCIR-10	TREC 2012
1	73.4%	74.3%	77.3%
2	15.7%	15.1%	12.7%
3	6.0%	5.7%	4.8%
4	2.7%	2.6%	2.5%
> 4	2.2%	2.3%	2.7%

Table 3: The distribution of documents relevant to different numbers of subtopics in NTCIR-9/10 Intent tasks and TREC 2012 Web track diversity datasets.

#(subtopic retrieved)	NTCIR-9	NTCIR-10	TREC 2012
1	48.1%	61.7%	56.0%
2	27.7%	24.3%	27.5%
3	13.4%	9.1%	10.1%
4	6.0%	3.4%	4.9%
> 4	4.8%	1.5%	1.5%

Definition 4.1. Denote by $rel_{k,i}$ $(0 < rel_{k,i} \leq 1)$ the relevance score for document d_k in subtopic $s_i \in C$. Two documents d_1 and d_2 are called an ordered pair (OP) if $rel_{1,i} > rel_{2,i}$ for any $s_i \in C$, denoted as $\langle d_1 \mapsto d_2 \rangle$.

To estimate the proportion of documents in which each document $\{d_i\}$ is relevant to a single subtopic, we analyze the datasets provided by NTCIR Intent tasks and TREC Web track diversity tasks and determine the number of subtopics to which a document can be relevant. We also look into the result lists for different subtopics retrieved by search systems to gain an insight into the number of subtopics in the retrieved documents.

Table 2 presents the percentages of documents that are retrieved for different numbers of subtopics in NTCIR-9/10 Intent and TREC 2012 Web track diversity tasks. For each subtopic, 1,000 documents were retrieved by our retrieval system with BM25 ranking. Meanwhile, because ideal lists are generated based on relevance judgment results, we also look into the statistics of the qrels (documents with relevance labeling) from these tasks as well. Table 3 shows the percentage of qrels that are relevant to different numbers of subtopics from these tasks. We can see from these two tables that for these collections, a majority of documents are relevant to only one subtopic both in candidate result documents and in annotated qrels. Documents that are relevant to only one subtopic are ordered, thereby forming a set of OPs. Thus, a large proportion of search branches corresponding to orders different from those of the OPs can be cut off in the pruning process of exhaustive search without compromising performance. Consider a query topic with n subtopics and a collection of N relevant documents with at least one subtopic annotated with a nonzero relevance score, and we denote the percentage of documents with k nonzero subtopic relevance scores by p_k for k from 1 to n. In the following theorem, we quantify the interplay between the number of OPs and the amount of subtopic coverage.

Theorem 4.2. *Let relevance scores be* i. i. d. *random variables with absolutely continuous density with respect to Lebesgue measure, and the distribution of k nonzero subtopics be uniform among the $\binom{n}{k}$ combinations; the probability that two documents form an OP is*

$$\mathbb{P}[\mathrm{OP}] = \sum_{k=1}^{n} \frac{p_k^2}{2^{k-1}\binom{n}{k}}. \tag{5}$$

Proof. Since the nonzero relevance scores are uniformly distributed among subtopics, the probability that two documents with k nonzero relevance scores have the same subtopics is $p_k^2 \Big/ \binom{n}{k}$. If the relevance scores of the documents are i. i. d. with absolutely continuous density, given two documents, the chance that one document has higher relevance scores for all k nonzero relevance scores than the other document is $\frac{1}{2^{k-1}}$, which concludes Equation (5). □

Given that most documents only cover one or two subtopics, Theorem 4.2 suggests that narrow subtopic coverage leads to more OPs, implying that the pruning algorithm is effective.

5 PRUNED EXHAUSTIVE SEARCH

In this section, we prove criteria to prune branches without compromising performance, and propose a pruning strategy for a more efficient exhaustive search to obtain the optimal diversified ranking by skipping useless branches.

5.1 Result Clustering to Find Ordered Pairs

While performing an exhaustive search, the candidates in a certain iteration include all the documents except those already selected. As discussed in Section 4, the candidate result documents may be relevant to different subtopics, and only those documents that are relevant to the same group of subtopics form OPs. Therefore, we group documents into different clusters by subtopics. For a certain subset of subtopics C', we group all candidate documents that are relevant to, and only to, all subtopics in C' into a cluster C. Each candidate is assigned to exactly one cluster, and can only form an OP with other candidates in the same cluster. The total number of clusters generated by a subtopic set C is:

$$\sum_{k=0}^{|C|} \binom{|C|}{k} = 2^{|C|}.$$

According to Tables 2 and 3, the percentage of documents relevant to more than four subtopics is expected to be less than 5%. Therefore, in actual Web search environments, the number of clusters should be less than $2^{|C|}$. Within each cluster, the candidate documents may form a number of OPs. In particular, for those clusters corresponding to only one subtopic, each pair of documents forms an OP. For other clusters, although not all pairs of documents form OPs, finding OPs is efficient since judging whether two documents form an OP only requires comparing their relevance scores for the subtopics involved.

5.2 Candidate Selection for Pruned Search

After clustering candidate documents according to the subtopics they are relevant to and identifying OPs from the clusters, we show in Theorems 5.1 and 5.2 that OPs could be adopted to reduce the number of candidates in each iteration of the exhaustive search.

Theorem 5.1. *Let $decay_i(S')$ be a function of the number of documents in S' that are relevant to s_i. Given an ordered pair $\langle d_1 \mapsto d_2\rangle$ of a subset of subtopics C', if only one document from the set $\{d_1, d_2\}$ appears in the optimal ranking list, that document should be d_1.*

Proof. Consider any list S containing d_1 and a copy of the list $\tilde{S}$ with d_1 replaced by d_2. Then

$$Score(S) = \sum_{l=1}^{|S|} G(d_l, l, S'_l),$$

$$Score(\tilde{S}) = \sum_{l=1}^{|S|} G(d_l, l, \tilde{S}'_l).$$

Since both d_1 and d_2 are relevant to the same subset of subtopics, switching d_1 and d_2 will not influence the decay scores of documents ranked lower than the l-th position by hypothesis. Therefore, $g_i(d)$ is the only element different between $G(d_1, l, S')$ and $G(d_2, l, S')$ and we have $g_i(d_1) \geq g_i(d_2)$ for any s_i, according to the definition of an ordered pair; hence, $Score(S) \geq Score(\tilde{S})$. □

Theorem 5.2. *Let $decay_i(S')$ be a monotonically decreasing function of the number of documents in S' that are relevant to s_i, and let $decay(\cdot)$ be monotonically decreasing. Given an ordered pair $\langle d_1 \mapsto d_2\rangle$ of a subset of subtopics C', if both d_1 and d_2 appear in the optimal ranking list, d_1 should be ranked higher than d_2.*

Proof. Let S_1 be a result list containing an ordered pair $\langle d_1 \mapsto d_2\rangle$, where d_1 is ranked at the k-th location, and d_2 is ranked at the l-th location with $k > l$, as illustrated in Figure 1. Switching the location of d_1 and d_2 yields a new result list, and we call it S_2. We show that $Score(S_1) > Score(S_2)$. Let us denote by S' the sub-list ranked higher than the l-th location, which is identical in S_1 and S_2. We use $S_{1,k}$ and $S_{2,k}$ for the sub-lists of documents ranked before the k-th position in S_1 and S_2 respectively; indistinguishably denoted as $S_{*,k}$. Then the diversified gain of d_1 and d_2 in S_1 is:

$$\begin{aligned} G(d_1, l, S') + G(d_2, k, S_{1,k}) = &\sum_{s_i \in C'} w_i \cdot g_i(d_1) \cdot decay_i(S') \cdot decay(l) \\ &+ \sum_{s_i \in C'} w_i \cdot g_i(d_2) \cdot decay_i(S_{1,k}) \cdot decay(k) \end{aligned}$$

The diversified gain of d_1 and d_2 in S_2 is:

$$\begin{aligned} G(d_2, l, S') + G(d_1, k, S_{2,k}) = &\sum_{s_i \in C'} w_i \cdot g_i(d_2) \cdot decay_i(S') \cdot decay(l) \\ &+ \sum_{s_i \in C'} w_i \cdot g_i(d_1) \cdot decay_i(S_{2,k}) \cdot decay(k) \end{aligned}$$

Since both d_1 and d_2 are relevant to the same subtopics, and $decay_i(S')$ is only related to the number of documents in S', for each subtopic $s_i \in C'$, $decay_i(S_{1,k}) = decay_i(S_{2,k})$. The difference between $Score(S_1)$ and $Score(S_2)$ is contributed by the diversified gain of d_1 and d_2:

$$\begin{aligned} &G(d_1, l, S') + G(d_2, k, S_{1,k}) - G(d_2, l, S') - G(d_1, k, S_{2,k}) \qquad (6) \\ &= \sum_{s_i \in C'} w_i \cdot g_i(\Delta d) \cdot (decay_i(S') \cdot decay(l) - decay_i(S_{*,k}) \cdot decay(k)) \end{aligned}$$

where $g_i(\Delta d) = (g_i(d_1) - g_i(d_2)) > 0$ by Definition 4.1. Since the decay functions $decay_i(\cdot)$ and $decay(\cdot)$ are monotonically decreasing, with the conditions $l < k$ and $S' \subset S_{*,k}$ we get:

$$\sum_{s_i \in C'} (decay_i(S') \cdot decay(l) - decay_i(S_2') \cdot decay(k)) > 0$$

Therefore, the value of Equation (6) is greater than 0, which concludes that $Score(S_1) > Score(S_2)$. □

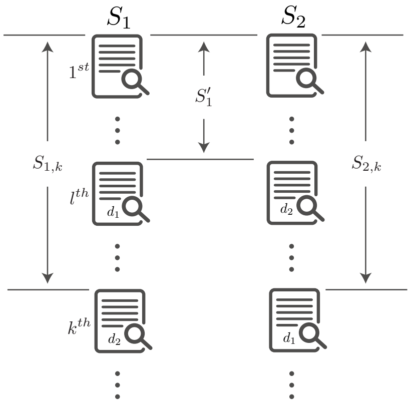

Figure 1: Result list S_1 and S_2.

Theorems 5.1 and 5.2 show that if we group documents into clusters of OPs, we can prune branches in the exhaustive search because the order of documents in an OP should be preserved in the optimal result ranking list. That is, if $\langle d_1 \mapsto d_2 \rangle$, d_1 should be selected as a candidate instead of d_2. Therefore, we propose the following PesOP algorithm (Algorithm 2) to prune the candidate selection process in each exhaustive search iteration. The function *select_candidates* generates the set of candidates for a position in an iteration, where each element in the set appears first in all of the OPs containing the element. When a document d from the candidate set is selected for the current position, it is removed from its cluster. Since the candidate selection process in Algorithm 2 only removes the branches that would not lead to the optimal ranking, the PesOP algorithm generates the optimal results as the original exhaustive search algorithm, but with improved efficiency.

6 EXPERIMENTS AND DISCUSSIONS

In the experiments, we answer the following research questions:

RQ1. Does the PesOP algorithm outperform the greedy search?

RQ2. How efficient is the PesOP algorithm compared with the exhaustive search and greedy search?

6.1 Experiment Setups

To answer the above research questions, we collected subtopics submitted in subtopic mining tasks. In the NTCIR-9 INTENT task and NTCIR-10 INTENT-2 task, participants were provided with 100 query topics in each task to mine the subtopics weighted by the importance estimations for each query topic. These tasks were performed by 18 teams, totaling 3,600 query instances. A text retrieval system retrieved initial lists for the top ten subtopics of each instance based on SogouT[1]. In addition to these 200 Chinese queries, 50 English queries from TREC 2012 Web track diversity task were performed by 29 teams independently, totaling 1,450 English query instances. Initial lists for the top ten subtopics of each instance were retrieved for the top ten subtopics based on ClueWeb09[2].

Algorithm 2: Pruned exhaustive search based on ordered pairs

```
Input: The set of initial documents $D$.
$cluster\_set = \emptyset$
for $d$ in $D$ do
    $I_d$ = the index set of intents with $rel_i(d) > 0$
    if $I_d \notin cluster\_set$ then
        $cluster\_set \leftarrow cluster\_set \cup I_d$
    end
    $cluster\_set(I_d) \leftarrow \{d\}$
end

Function select_candidates()
    $candidates = \emptyset$
    for each $I_d$ in $cluster\_set$ do
        $OP\_set(I_d)$ = all OPs in $cluster\_set(I_d)$
        for each $d$ in $cluster\_set(I_d)$ do
            if $d$ is never the second element in $OP\_set(I_d)$ then
                $candidates \leftarrow candidates \cup \{d\}$
            end
        end
    end
return $candidates$

Function Recursion($S$, $D$, $returnS$)
    if $S = |L|$ then
        $returnS \leftarrow returnS \cup S$
        return $Score(returnS)$
    end
    else
        for each $d$ in $D$ do
            $S \leftarrow S \cup \{d\}$
            return $Recursion(S, D \setminus \{d\}, returnS)$
        end
    end

$maxG = 0$
$S \leftarrow \emptyset$
for each $d$ in $select\_candidates$ do
    $returnS \leftarrow \emptyset$
    $curG = Recursion(\{d\}, select\_candidates \setminus \{d\}, returnS)$
    if $maxG < curG$ then
        $maxG \leftarrow curG$
        $S \leftarrow returnS$
    end
end
return $S$
```

[1] https://www.sogou.com/labs/resource/t.php

[2] http://lemurproject.org/clueweb09/

Table 4: Percentage of optimal lists generated by IA-Select [1], Utility [2], xQuAD [27], CombSUM [19], PM2 [10], and PesOP for different queries.

Dataset	Algorithm	Required Result List Length L			
		2	3	4	5
Chinese Queries (NTCIR -9&10)	PesOP	100%	100%	100%	100%
	IA-Select	94.0%	86.9%	79.6%	71.9%
	Utility	0.10%	0.03%	0.00%	0.00%
	xQuAD	1.04%	0.42%	0.39%	0.39%
	CombSUM	1.06%	0.43%	0.39%	0.37%
	PM2	17.5%	8.94%	4.96%	2.69%
English Queries (TREC 2012)	PesOP	100%	100%	100%	100%
	IA-Select	92.8%	82.2%	71.0%	59.1%
	Utility	0.00%	0.00%	0.00%	0.00%
	xQuAD	2.78%	0.42%	0.00%	0.00%
	CombSUM	2.75%	0.45%	0.00%	0.00%
	PM2	21.60%	8.54%	3.61%	1.74%

The top 20 documents are labeled with relevance scores. We normalize the relevance score of each document by the maximum score of documents in the same subtopic, which is taken as $P(d|q, s_i)$ in Equation (3). The subtopics with their weights and their retrieved results are taken as the input of all diversified search algorithms (Algorithms 1, 2) to determine which method produces diversified result rankings with higher revised α-NDCG as described in Equation (3), which takes the factor of $decay_i(S')$ into consideration. The revised α-NDCG is adopted instead of its original version because the original α-NDCG assumes that only binary relevance judgment is available and all subtopics are equally important. Since the most recent benchmark datasets such as TREC Diversity and NTCIR Intent tasks provide multi-grade relevance judgment and subtopic importance estimations, it is natural to extend the measure to fit such graded relevance judgments.

6.2 Diversified Result Ranking

With the above experimental settings, we first try to answer the research question RQ1 by investigating how the proposed algorithms work in the diversified ranking task. Since it is costly for the exhaustive search to generate a diversified ranking with $L > 5$, we limit the required list length to 2, 3, 4 and 5 to compare the efficiency of the algorithms. Table 4 shows the percentage of query instances for which each algorithm generates optimal ranking lists. The performance of exhaustive search, greedy search (Algorithm 1, with IA-Select [1] as an example), Utility method [2], xQuAD [27], CombSUM [19], PM2 [10], and PesOP (Algorithm 2) are compared while they are performed on the results retrieved from NTCIR and TREC datasets with our retrieval system.

From Table 4 we can see that with different L and different datasets, the PesOP algorithm generates the optimal results just as the original exhaustive search algorithm does. This agrees with Theorems 5.1 and 5.2, which ensure that removing branches that lead to result lists against existing ordered pairs will not affect the optimality of exhaustive search algorithms. Different from the proposed PesOP algorithm, IA-Select [1], Utility [2], xQuAD [27], CombSUM [19], and PM2 [10] fail to obtain optimal results for some cases according to Table 4. IA-Select obtains better results than the other baselines because the diversified search results are evaluated based on the revised α-NDCG. While selecting a document for the ranking list, IA-Select always chooses the candidate with the largest α-NDCG value in each step, whereas the other methods choose documents according to other standards, which may be much different from α-NDCG. Therefore, few of their results can be optimal when evaluated by the revised α-NDCG. Table 4 also shows that when the length of the required result list increases, the percentage of optimal ranking for IA-Select drops from 94% to 71.9% on Chinese datasets and from 92.8% to 59.1% on English datasets. This reflects the fact that the greedy search targets local optimality. In fact, the xQuAD, CombSUM, and PM2 are also greedy algorithms, but their different evaluation metrics make them hardly comparable with the greedy search and PesOP.

Table 5: Average time cost per query (in seconds) for diversified document ranking with different retrieval length.

Dataset	Algorithm	Required Result List Length L			
		2	3	4	5
NTCIR -9&10	Exhaustive Search	0.0260	4.2660	724.4	81283
	Greedy Search	0.0003	0.0004	0.0004	0.0005
	PesOP	0.0018	0.0229	0.6166	19.95
TREC 2012	Exhaustive Search	0.0224	3.715	549.5	32359
	Greedy Search	0.0003	0.0003	0.0004	0.0005
	PesOP	0.0019	0.0251	0.6918	22.91

6.3 Efficiency of Algorithms

We investigate the time cost of the proposed algorithm and compare with the exhaustive search and greedy search. Table 5 presents the time cost of the experiments in seconds, as discussed in Section 6.2. All experiments are performed on a Linux server with a 12-core AMD Opteron CPU and 64-Gigabyte memory.

Table 5 shows that the time cost of the original exhaustive search increases quickly as L increases, making it infeasible to obtain a ranking list for larger L. On the other hand, the greedy search runs approximately in linear time. This makes it particularly suitable for situations in which a fast selection of diversified result lists is required. Although the time cost of the PesOP algorithm is higher than that of the greedy algorithm, it is much lower than that of the exhaustive search, and provides the same ranking performance as exhaustive search according to Table 4.

7 CONCLUSION

The problems of generating diversified search results and ideal ranking lists are two important issues in Web search diversification. They have been proven to be NP-Hard, and solutions based on the exhaustive search are impractical. Greedy search strategies have been adopted in prior studies, but the results are non-optimal. In this paper, we propose a pruned exhaustive search algorithm (PesOP) to reduce the complexity of exhaustive search and to generate the optimal ranking list based on findings in diversified search data sets. Pruning strategies are developed based on ordered pairs (OPs), according to which a large number of candidates generated in each exhaustive search iteration can be reduced. Experimental results based on NTCIR and TREC datasets showed that the revised PesOP algorithm significantly outperforms the widely used greedy search algorithm IA-Select [1], Utility [2], xQuAD [27], CombSUM [19], and PM2 [10]. The approaches proposed in this paper represent only the first step towards better result rankings and more reliable system evaluations in the diversified search. Many questions remain to be addressed, such as how to develop more efficient methods when reasonable relaxation on the optimality is allowed, and how to reduce the time complexity of determining the optimal ranking for the top ten results and beyond.

8 ACKNOWLEDGEMENTS

The authors thank Dr. Andrew Blumberg, Dr. Kui Ren and Dr. François Baccelli at the University of Texas at Austin, Dr. Hao Wu at Tsinghua University, and Dr. Lizhen Ji at the University of Michigan for suggestions on mathematical proofs. This work is supported by the Natural Science Foundation of China (Grant No. 61622208, 61732008, 61532011) and the National Key Basic Research Program (2015CB358700). Wu was supported in part by NIH grant 5U54CA193313.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the 2nd International Conference on Web Search and Data Mining*. 5–14.

[2] Gabriele Capannini, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. 2011. Efficient diversification of web search results. *Proceedings of the VLDB Endowment* 4, 7 (2011), 451–459.

[3] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[4] Ben Carterette. 2011. An analysis of NP-completeness in novelty and diversity ranking. *Information Retrieval* 14, 1 (2011), 89–106.

[5] Ben Carterette and Praveen Chandar. 2009. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 1287–1296.

[6] Olivier Chapelle, Shihao Ji, Ciya Liao, Emre Velipasaoglu, Larry Lai, and Su-Lin Wu. 2011. Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval* 14, 6 (2011), 572–592.

[7] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 621–630.

[8] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st ACM SIGIR conference on Research and development in information retrieval*. 659–666.

[9] Charles LA Clarke, Maheedhar Kolla, and Olga Vechtomova. 2009. An effectiveness measure for ambiguous and underspecified queries. In *Proceedings of the 2nd ACM SIGIR International Conference on Theory of Information Retrieval*. 188–199.

[10] Van Dang and W Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th ACM SIGIR conference on Research and development in information retrieval*. 65–74.

[11] Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. 2011. Multi-dimensional search result diversification. In *Proceedings of the 4th International Conference on Web Search and Data Mining*. 475–484.

[12] Marina Drosou and Evaggelia Pitoura. 2012. Dynamic diversification of continuous data. In *Proceedings of the 15th International Conference on Extending Database Technology*. ACM, 216–227.

[13] Veronica Gil-Costa, Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2013. Modelling efficient novelty-based search result diversification in metric spaces. *Journal of Discrete Algorithms* 18 (2013), 75–88.

[14] Jiyin He, Edgar Meij, and Maarten de Rijke. 2011. Result diversification based on query-specific cluster ranking. *Journal of the Association for Information Science and Technology* 62, 3 (2011), 550–571.

[15] Neil Hurley and Mi Zhang. 2011. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)* 10, 4 (2011), 14.

[16] Qiaozhu Mei, Jian Guo, and Dragomir Radev. 2010. Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. Acm, 1009–1018.

[17] David R Morrison, Sheldon H Jacobson, Jason J Sauppe, and Edward C Sewell. 2016. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization* 19 (2016), 79–102.

[18] Lukas Neumann and Jiri Matas. 2011. Text localization in real-world images using efficiently pruned exhaustive search. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 687–691.

[19] Ahmet Murat Ozdemiray and Ismail Sengor Altingovde. 2015. Explicit search result diversification using score and rank aggregation methods. *Journal of the Association for Information Science and Technology* 66, 6 (2015), 1212–1228.

[20] Judea Pearl. 1980. SCOUT: A Simple Game-Searching Algorithm with Proven Optimal Properties.. In *AAAI*. 143–145.

[21] Judea Pearl. 1982. The solution for the branching factor of the alpha-beta pruning algorithm and its optimality. *Commun. ACM* 25, 8 (1982), 559–564.

[22] Aske Plaat, Jonathan Schaeffer, Wim Pijls, and Arie De Bruin. 1996. Best-first fixed-depth minimax algorithms. *Artificial Intelligence* 87, 1-2 (1996), 255–293.

[23] Davood Rafiei, Krishna Bharat, and Anand Shukla. 2010. Diversifying web search results. In *Proceedings of the 19th international conference on World Wide Web*. ACM, 781–790.

[24] Marco Tulio Ribeiro, Anisio Lacerda, Adriano Veloso, and Nivio Ziviani. 2012. Pareto-efficient hybridization for multi-objective recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 19–26.

[25] Tetsuya Sakai. 2012. Evaluation with informational and navigational intents. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 499–508.

[26] Tetsuya Sakai, Zhicheng Dou, Takehiro Yamamoto, Yiqun Liu, Min Zhang, Ruihua Song, MP Kato, and M Iwata. 2013. Overview of the NTCIR-10 INTENT-2 Task.. In *NTCIR*.

[27] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World Wide Web*. ACM, 881–890.

[28] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2010. Selectively diversifying web search results. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 1179–1188.

[29] Rodrygo LT Santos, Craig Macdonald, Iadh Ounis, et al. 2015. Search result diversification. *Foundations and Trends® in Information Retrieval* 9, 1 (2015), 1–90.

[30] Ruihua Song, Zhicheng Dou, Hsiao-Wuen Hon, and Yong Yu. 2010. Learning query ambiguity models by using search logs. *Journal of Computer Science and Technology* 25, 4 (2010), 728–738.

[31] Ruihua Song, Min Zhang, Tetsuya Sakai, Makoto P Kato, Yiqun Liu, Miho Sugimoto, Qinglei Wang, and Naoki Orii. 2011. Overview of the NTCIR-9 INTENT Task.. In *NTCIR*. Citeseer.

[32] Jun Wang and Jianhan Zhu. 2009. Portfolio theory of information retrieval. In *Proceedings of the 32nd ACM SIGIR conference on Research and development in information retrieval*. 115–122.

[33] Yingying Wu, Yiqun Liu, Ke Zhou, Xiaochuan Wang, Min Zhang, and Shaoping Ma. 2018. Treating Each Intent Equally: The Equilibrium of IA-Select. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 113–114.

[34] Yufei Xue, Fei Chen, Aymeric Damien, Cheng Luo, Xin Li, Shuai Huo, Min Zhang, Yiqun Liu, and Shaoping Ma. 2013. THUIR at NTCIR-10 INTENT-2 Task. In *NTCIR*.

[35] Yufei Xue, Fei Chen, Tong Zhu, Chao Wang, Zhichao Li, Yiqun Liu, Min Zhang, Yijiang Jin, and Shaoping Ma. 2011. THUIR at NTCIR-9 INTENT Task. In *NTCIR*. Citeseer.

[36] Cong Yu, Laks Lakshmanan, and Sihem Amer-Yahia. 2009. It takes variety to make a world: diversification in recommender systems. In *Proceedings of the 12th international conference on extending database technology: Advances in database technology*. ACM, 368–378.

[37] Long Yuan, Lu Qin, Xuemin Lin, Lijun Chang, and Wenjie Zhang. 2016. Diversified top-k clique search. *The VLDB JournalâĂŤThe International Journal on Very Large Data Bases* 25, 2 (2016), 171–196.

[38] Min Zhang, Chuan Lin, Yiqun Liu, Leo Zhao, and Shaoping Ma. 2003. THUIR at TREC 2003: Novelty, Robust and Web. In *TREC*. 556–567.

[39] Le Zhao, Min Zhang, and Shaoping Ma. 2006. The nature of novelty detection. *Information Retrieval* 9, 5 (2006), 521–541.

[40] Guido Zuccon, Leif Azzopardi, Dell Zhang, and Jun Wang. 2012. Top-k retrieval using facility location analysis. In *European Conference on Information Retrieval*. Springer, 305–316.