SEARCH RESULT DIVERSIFICATION FOR SELECTIVE SEARCH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE CAN KÜÇÜKOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

DECEMBER 2019

Approval of the thesis:

SEARCH RESULT DIVERSIFICATION FOR SELECTIVE SEARCH

submitted by **EMRE CAN KÜÇÜKOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. İsmail Sengör Altıngövde
Supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering Dept., METU

Assoc. Prof. Dr. İsmail Sengör Altıngövde
Computer Engineering Dept., METU

Assist. Prof. Dr. Engin Demir
Computer Engineering Dept., Hacettepe University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Emre Can Küçükoğlu

Signature :

ABSTRACT

SEARCH RESULT DIVERSIFICATION FOR SELECTIVE SEARCH

Küçükoğlu, Emre Can
M.S., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. İsmail Sengör Altıngövde

December 2019, 44 pages

Our work explores the performance of result diversification methods in the selective search scenario, where the underlying document collection is topically partitioned across several nodes and the search is conducted only at a subset of these nodes. In particular, we investigate whether diversification at each node is superior to previous approaches in the literature, i.e., diversification at the broker node applied before the resource selection or after the result merging stages. We also compare performance of different centralized sample indexes to show their effect on diversification. Finally, we explore the impact of recently introduced query expansion techniques using word embeddings to improve the effectiveness of diversification applied at the broker node, and subsequently, overall diversification. Our experiments reveal that for implicit diversification methods, expanding queries with diversified terms and applying diversification during the resource selection stage yield the best performance. In contrary, for explicit diversification methods, diversifying merged results at the broker is the best solution.

Keywords: Search Result Diversification, Selective Search, Distributed Search, Query Expansion, Word Embeddings

ÖZ

SEÇMELİ ARAMA İÇİN ARAMA SONUCUNU ÇEŞİTLENDİRME YÖNTEMLERİ

Küçükoğlu, Emre Can
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. İsmail Sengör Altıngövde

Aralık 2019 , 44 sayfa

Bu tezde, konularına göre gruplandırılmış, büyük doküman kümelerinin, seçmeli arama yöntemleriyle sorgulanması sürecindeki, sonuç çeşitlendirme tekniklerinin performansları incelenmiştir. Çeşitlendirme yöntemleri, seçmeli aramanın farklı katmanlarına uygulandığında sonuç üzerinde gösterdiği etkiler de çeşitlilik göstermektedir. Bu tezde, dağıtıcı üzerindeki sonuç listelerinin çeşitlendirilmesinin; kaynak seçimi öncesi ve sonrası uygulanmasının genel sonuç çeşitliliğine etkileri incelenmiştir. Buna ek olarak, farklı merkezi indekslerin, sonuç çeşitlendirme performansına etkileri incelenmiştir. Son olarak, güncel bir konu olan, kelime vektörleriyle sorgu genişletmenin, seçmeli arama sistemlerindeki sonuç çeşitlendirme üzerine etkisi ortaya konulmuştur. Çalışmalarımız göstermiştir ki, kapalı anlam sonuç çeşitlendirme yöntemlerinde, sorgu kelimelerinin çeşitlendirilerek genişletilmesi ve çeşitlendirmenin kaynak seçimi aşamasında uygulandıgı teknikler en iyi performansı vermektedir. Açık anlam sonuç çeşitlendirme yöntemlerinde ise, dağıtıcının birleştirilmiş sonuç listelerinde yapılan çeşitlendirme yöntemleri en iyi sonuçları üretmektedir.

Anahtar Kelimeler: Arama Sonucu Çeşitlendirme, Seçmeli Arama, Dağıtık Arama, Sorgu Genişletme, Kelime Vektörleri

To my family and friends...

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Assoc. Prof. Dr. İsmail Sengör Altıngövde for his guidance and everything he taught me. It was a great opportunity to work with him, and I feel extremely lucky for being one of his students. He always encouraged me to do better and thanks to his constructive criticism I was able to complete this work.

Thanks to Sengör hoca, I had a chance to work with various talented colleagues, I would like to thank Andaç Akarsu and Yaşar Barış Ulu.

And, my dear friend, Fatih Hafızoğlu, the best co-worker I have ever had. I thank him for everything we shared together and wish him success and happiness in all his life.

Also, I would like to express my gratitude to my family who always supports me at times I need them. I could not be who I am without them, thank you my mother, father, my lovely sisters. I love you all.

Lastly, I owe a debt of gratitude to my friends in Ankara; Ezgi, Necati, Alper, Ugur and Dokuz. I am so glad to know you all.

This work is partially funded by the Ministry of Science, Industry & Technology of Turkey and Huawei Inc. under the grant no 0441.STZ.2013-2 and also by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant no. 117E861.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

BM25	Best Matching - 25
CSI	Centralized Sample Index
IR	Information Retrieval
RS	Resource Selection
GAVG	Geometric Average based Collection Selection
REDDE	Relevant Document Distribution Estimation
CRCSExp	Exponential Central-Rank-Based Collection Selection
CRCSLin	Linear Central-Rank-Based Collection Selection
MaxSum	Max-Sum Dispersion Diversification
MMR	Maximal Marginal Relevance
MMRE	MMR-based Expansion
xQuAD	Explicit Query Aspect Diversification
DDiv	Diversification approach based on sample Documents
α-nDCG	α Weighted Normalized Discounted Cumulative Gain

CHAPTER 1

INTRODUCTION

1.1 Motivation

Searching in a collection of documents is done using indexes of these documents. The simplest approach is that the single index stores information for every single document, then search query can be run over this index. Apparently, this basic approach is not scalable, as the index grows linearly proportional to the number of documents. Distributing the documents across separate nodes allows conducting search in parallel. In this distributed approach, there is a broker node that sends search queries to these nodes with the documents and the corresponding indexes, and then gather the results from them. Broker sends the query to all nodes without making any selection. Therefore, all documents are still be processed by the query. This approach is called as exhaustive search on a distributed architecture.

Allocating the documents in different nodes can be done in various ways. For example, documents can be grouped by their topical similarity, therefore we can say that related documents will be all together. It is also possible to allocate them according to their source information, i.e. URL information for web based search. If we distribute the documents to the nodes using such representative information, i.e., store *topical clusters* at each node; then at the broker node, we can estimate which clusters may yield better results for the query (called as the resource selection stage [1]). To do that, broker stores information about all the clusters, which is called as centralized sample index, CSI. This technique, called as selective search [1], does not process all the documents, however it still can be as effective as the exhaustive search.

In search concept, main purpose is to satisfy the user's expectation, which naturally involves a diversity among different needs. To achieve this, different aspects of the query should be taken into consideration before returning results. However, queries aren't always clear enough. Because of the ambiguity of the queries, it is difficult to understand user's actual intention. Search engines use the similarities between the documents and the query for retrieval. However, this approach may lead to result lists that include only similar documents. Therefore, there should be an additional procedure to diversify the results to cover different possible user intentions. On the other hand, there is a trade-off between diversity and relevance: covering more aspects than the user's need can cause including irrelevant documents.

For example, if we consider the web search engines, many users type only couple of words to find the information they are looking for. A user who wants to find something about fruit apple, may forget to add word 'fruit' to query, because of that, most probably the user will be misled to 'Apple the Company'. The lack of more explanatory words like 'fruit', make it difficult to match user expectations. For this kind of ambiguous queries, search engines are expected to generate a result set which includes different aspects of the same query. Despite the fact that they should consider query ambiguities, the trade-off between diversity and relevance prevents them to add documents randomly to the result set just because they improve the diversity of the set [2].

For selective search, there are several layers where diversification can be applied. For example, at the broker node, diversification can be applied just before the result set is being returned, or each cluster may try to diversify their partial result set before sending them to broker. In all of these cases; search result diversification can be done either in an explicit way, where search engines have an external knowledge about queries; or implicitly (as will be discussed later). For both cases, the main purpose is to maximize the coverage to solve ambiguity issues and minimize redundancy, while keeping relevance still high.

1.2 Contribution of Thesis

In this thesis, we contribute the literature in several ways, listed as follows.

- We propose alternative ways of creating a centralized sample index (CSI) based on various document properties and evaluate their impact on the resource selection and subsequently, overall diversification effectiveness.
- We expand queries using word embeddings and apply diversification to obtain diverse expansion terms, as in [3]. As a novel contribution, we process such expanded queries on the CSI, i.e., during the resource selection stage, to obtain more diverse resources.
- For selective search, we explore the performance of search result diversification at different layers (i.e., at the broker vs. in the clusters) and at different stages, namely, before resource selection and before/after result merging. To the best of our knowledge, while diversification performance is investigated over a distributed setting for exhaustive search (i.e., with random assignment of documents to nodes), our work is the first to make such a detailed analysis in the context of selective search and by employing representative strategies for both implicit and explicit search result diversification.

1.3 Organization of Thesis

In Chapter 2, we will review the studies in the literature for the result diversification for selective search. This chapter will give background information about algorithms and approaches we adapted for our experiments; i.e., well-known implicit and explicit diversification algorithms, resource selection, query expansion and document allocation techniques used in our experiments. In Chapter 3, we will explain the techniques that are used to create different CSIs and also how we expand the queries using word embeddings. In Chapter 4 we will present different approaches of diversification for selective search on which we build up our experiments. Then, respectively in Chapters 5 and 6, detailed information about experiments will be presented. Finally, in Chapter 7, we will conclude our work with a summary and possible future work.

CHAPTER 2

RELATED WORK

In this chapter, firstly we will present selective search and its document allocation and CSI generation steps that we also adapted for our work. We will also explain different studies for resource selection phase of selective search. Then, we will present various diversification approaches studied in the literature so far. Finally, we will discuss word embeddings and its usage to extend queries.

2.1 Selective Search

Selective search is an approach to search in very large scale environment. Partitioned documents are splitted into clusters. In the center of the selective search, there is a broker which technically manages the process. It sends the query to the clusters. In Figure 2.1, broker is responsible to send the query to selected clusters, that are 1 and 3. However, to decide which clusters to send, it first runs the query in centralized sampled index. After sending the query to selected clusters, broker then collects and merges the results. This technique, which was previously known as a cluster based retrieval [4] aims to reduce resource usages while keeping the accuracy of the results.

Kulkarni and Callan [1]'s studies show that selective search approach can be used instead of exhaustive search, since overall effectiveness of the resultant sets are not so different. Main advantage and purpose of the selective search is reducing the overall time spent by the search engine. In next sections, we will respectively take a look at the studies in clustering the documents, choosing a centralized index for them, and finally selection techniques over these centralized sample indexes.

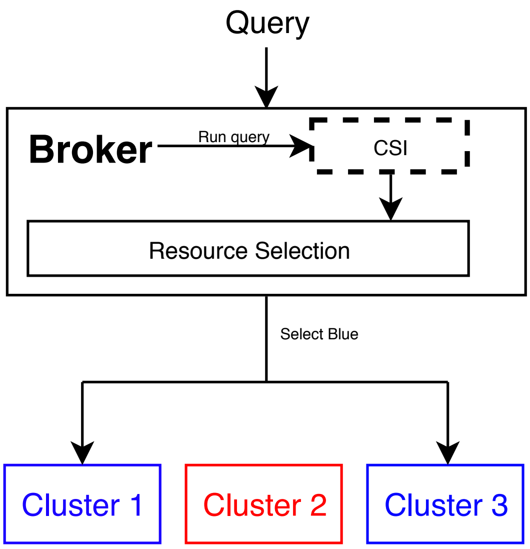


Figure 2.1: Selective search over partitioned clusters using CSI.

2.1.1 Document Allocation Policies

For selective search, each physical shards consist of collection of documents. The main issue here is to determine the relation between documents in same shards. These allocations can be done using any kind of information of documents. Currently, there are three different types of document allocation policies studied, these are random selection, source based selection and topic based selection [5]. Random based allocation choose documents randomly. Source based is mainly depends on the url addresses of web documents.

In this thesis, we applied topically partitioned shards using Kullback-Liebler divergence method that is introduced by Xu and Croft [6]. As shown in Algorithm 1 and in Figure 2.2; after documents in the collection are sampled, these sampled documents are used to iteratively generate cluster centroids that is used to assign documents.

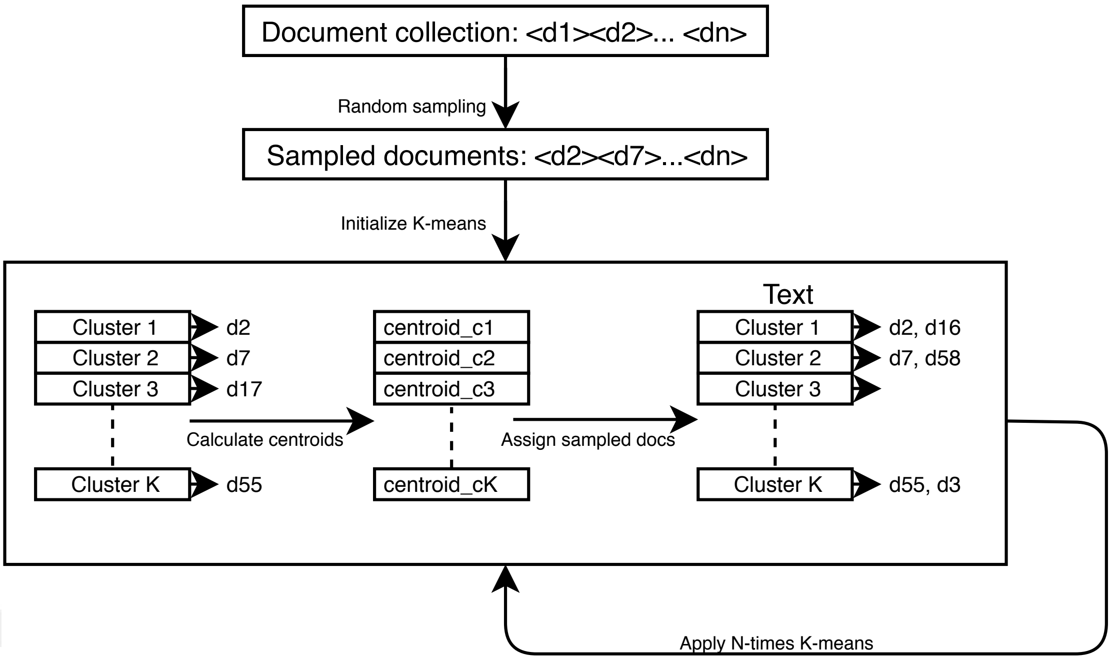


Figure 2.2: Topically Document Allocation simulation.

Algorithm 1 Topically Partitioning using K-means and Kullback-Leibler divergence

Input: Document collection C, Number of Clusters K, Times of K-means N, Sampling Percentage P

Output: Topical clusters CT

```
SAMPLEDOCS ← RandomSampleDocs(C, K, P)
$CT$ ← InitializeKMeans(SAMPLEDOCS, K)
CENTROIDS ← CalculateKLCentroids(RANDOMDOCS) // Kullback-Liebler divergence
**for** N Times **do**
    **for** Sampled Document $d \in C$ **do**
        **for** $k \in \{1, \dots, K\}$ **do**
            FIND $DISTANCE(CT_i, D)$
        **end for**
        Assign $d$ to $CT_n$ where $CT_n$ is the closest cluster
    **end for**
**end for**
**return** $CT$
```

2.1.2 Centralized Sample Index (CSI)

After documents are allocated different clusters, each one should be represented at broker in an index. In this index, instead of including information from all documents; it is possible to maintain selection quality at broker by including a subset of them. Current experiments in literature shows that this subset, namely centralized sample index can return effective results and represent the real clusters good enough [1].

Aly et al. [7] introduce a randomly generated centralized sample index approach. In this approach, each cluster will be represented equally at broker, because each one of clusters will send same percentage of documents that are randomly chosen. We adapted this approach to create CSI.

2.1.3 Resource Selection

Selective search is based on searching multiple clusters at the same time. To achieve that, subset of clusters are questioned by queries and selected clusters' results are merged into a single list [8].

Here the main issue is how to choose correct clusters. In the literature, resource selection is mainly divided into two main categories, big document approach and small document approach. Former one is the first generation of resource selection techniques. In this technique, resources are treated as a bag of words, which are concatenation of its documents or its sampled documents. Based on this, resource selection task is reduced to a document retrieval task [9].

Another resource selection model is the small document model, which ranks the sample documents according to occurrences of documents in the result set. In this thesis, we tried well known resource selection algorithms that apply small document model, these are Redde [10], ReddeTop [11], [12], CRCSLin, CRCSExp [13] and GAVG [14].

Redde resource selection method is proposed after noticing that big document approaches does not do well in environments that contain small and large documents. Instead of using meta documents, this approach uses the information about collection size and result list. Running query in sampled documents creates a ranked list and from this list, a score for resource is calculated by only looking the total number of documents from that resource in top documents. This score is the score that shows relevancy of resource and query but it is not the final score. For finalizing score, as shown in Equation 2.1, Redde uses the ratio of the size of resources sampled documents to the size of resource.

$$Score(C^i, q) = n \times \frac{|C^i|}{|S^i|} \tag{2.1}$$

where

- C^i is the cluster to score
- q is the query
- n is the number of documents in top results from C^i
- S^i is the set of sampled documents from C^i
- $|C^i|$ is the size of C^i, i.e. number of documents in C^i
- $|S^i|$ is the size of S^i, i.e. number of sampled documents from C^i

In ReddeTop, which is proposed by Arguello et al. [12], scores of the documents are used instead of their ranks, it is shown in Equation 2.2

$$Score(C^i, q) = \sum_{d \in S^i, d \in R} P(d, q) \times \frac{|C^i|}{|S^i|} \tag{2.2}$$

where

- R is the top N result obtained from CSI
- $P(d, q)$ is the score of document d in query q

In addition to Redde and ReddeTop; CRCSLin and CRCSExp [13] other methods that take into consider ranks of the documents in the result list. Their equations are shown in Equation 2.3 and Equation 2.4 The main difference between CRCSLin and

CRCSExp is that latter consider ranks exponentially, where the former has linear equation.

$$Score(C^i, q) = \sum_{d \in S^i, d \in R} n - r \times \frac{|C^i|}{|S^i|} \tag{2.3}$$

$$Score(C^i, q) = \sum_{d \in S^i, d \in R} \alpha \epsilon^{-\beta \times r} \times \frac{|C^i|}{|S^i|} \tag{2.4}$$

where

- r is the rank of document d
- α and β are the coefficients

GAVG is the another resource selection algorithm which uses document scores as a variable to determine best clusters [14]. As it is shown in Equation 2.5, it calculates geometric averages of the document scores.

$$Score(C^i, q) = (\prod_{d \in S^i, d \in \text{top m of} S^i \in R} P(d, q))^{\frac{1}{m}} \tag{2.5}$$

2.2 Diversification for Selective Search

For selective search, processed queries yield a result set which is a sorted document list with respect to their relevancies to the query. Search result diversification is the process of reordering of these result sets in a way that they will handle the ambiguity of the existing query. This problem is defined as a NP-hard problem [15].

Search result diversification methods are mainly divided into two categories, implicit and explicit techniques [16]. Implicit diversification methods don't need external knowledge, and mainly depend on relevance results of the documents. To cover other aspects of the query, it re-rank these lists.

Many implicit diversification techniques try to maximize the total score of the set by taking into account relevance and dissimilarity scores. For example, MaxSum dispersion algorithm [17], as shown in Algorithm 2, tries to maximize objective function, Equation 2.6. Requirement to have document vectors in the place that implicit algorithm is being applied, is one drawback of this method. It needs document vectors to calculate SIM and DIV functions.

$$f(d_i, d_j) = (1 - \lambda)(SIM(q, d_i) + SIM(q, d_j)) + 2\lambda DIV(d_i, d_j) \tag{2.6}$$

where

- d_i, d_j documents to compare
- q query
- λ trade-off variable between similarity and dissimilarity
- SIM similarity function
- DIV dissimilarity function

Algorithm 2 MaxSum Dispersion Algorithm

```
Input: Document set S, result set size k
Output: Re-ranked list R, |R| = k
$R \leftarrow$ InitializeEmptyResultList()
for $i \in \{1, \ldots, k/2\}$ do
    FIND $(d_i, d_j)$=$argmax_{d_i, d_j \in S}$ $(f(d_i, d_j))$
    SET $R \leftarrow R$ U $\{d_i, d_j\}$
    SET $S \leftarrow S \setminus \{d_i, d_j\}$
end for
if $k$ is odd then
    select an arbitrary document $d_i \in S$
    SET $R \leftarrow R$ U $\{d_i\}$
end if
return $R$
```

SY is another implicit diversification algorithm, that aims to find near duplicate documents in the collection set [18]. In a sorted result set, it computes document to document relevancy scores and removes the ones that have a score more than threshold, as shown in Algorithm 3.

Algorithm 3 SY Algorithm

```
Input: Document set $S$, result set size $k$, threshold $\lambda$
Output: Re-ranked list $R$, $|R| = k$
$R \leftarrow$ InitializeEmptyResultList()
$i \leftarrow 0$
while $i$ < $k$ and $i$ < $|S|$ do
    $j \leftarrow i + 1$
    while j < |S| do
        if $SIM(S[i], S[j]) > \lambda$ then
            REMOVE $S[j]$ from $S$
        else
            $j \leftarrow j + 1$
        end if
    end while
    SET $R \leftarrow R\ U\ S[i]$
    $i \leftarrow i + 1$
end while
return $R$
```

Unlike implicit algorithms, explicit search result diversification algorithms use an external knowledge, like subtopics of existing queries to cover different aspects. Query aspects are used to re-rank the candidate result list. xQuAD is one promising explicit diversification algorithm in the literature [19]. It tries to maximize its objective function, Equation 2.7, by diversifying the relevance scores between the original query and documents, using the relation between subtopics of the query and documents. The product term in the equation computes novelty values for each aspect. xQuAD algorithm is shown in Algorithm 4.

$$f_{xQuAD}(d_i) = (1-\lambda)P(d_i|q) + \lambda \sum_{q_i} \left(P(q_i|q)P(d_i|q_i) \prod_{d_j \in R} (1 - P(d_j|q_i)) \right) \tag{2.7}$$

where

- $P(d_i|q)$ is relevance of d_i to query q
- $P(q_i|q)$ is likelihood of the aspect q_i for query q
- $P(d_i|q_i)$ is relevance of d_i to the aspect q_i

Algorithm 4 xQuAD Algorithm

```
Input:  Document set $S$, result set size $k$, tradeoff $\lambda$
Output:  Re-ranked list $R$, $|R| = k$
$R \leftarrow$ InitializeEmptyResultList()
$i \leftarrow 0$
while $i < k$ and $i < |S|$ do
    FIND $d^* \leftarrow argmax_{d \in S}\ f_{xQuAD}(d)$
    $S \leftarrow S - d^*$
    $R \leftarrow R\ U\ d^*$
    $i \leftarrow i + 1$
end while
return $R$
```

In another study, Hong and Si [20] propose two alternative approaches for search result diversification techniques. First of them is diversifying the document ranking while selecting clusters. This approach basically applies various diversification algorithms to result set which gained by processing the query over centralized sample index, CSI. This algorithm, named as Diversification approach based on sample Documents (DDiv).

In the same study, Diversification approach based on Source-level estimation (DivS) is the other proposed diversification technique. It treats clusters like a single big document, and computes their probability of relevance of query. Main advantage is that DivS also naturally supports wider range of resource selection algorithms. On the other hand, it requires an additional computation power to compare scores of query aspects for each clusters.

Hong and Si [20] conclude their work that both DDiv and DivS can outperform traditional methods. They also showed that resource level result diversification over CSI results can outperform source level diversification techniques. In this thesis, we will adapt their work and apply DDiv algorithm to compare different applications of search result diversification.

2.3 MMRE: Diversified Query Expansion using Word Embeddings

In this thesis, we also adapted an approach from Bouchoucha et al. [3] to expand the query words. They use the Maximal Marginal Relevance (MMR) implicit diversification algorithm [21] for selecting expansion terms. Their algorithm, called as MMRE (MMR-based Expansion), shown in Algorithm 5, searches for best documents which maximize MMRE function in Equation 2.8.

$$f_{mmre}(w_i) = \lambda P(w_i|q) - (1-\lambda) MAX_{w^{‘} \in R}\left(P(w^{‘}|q)\right) \tag{2.8}$$

where

- $P(w_i|q)$ is relevance of w_i to query q
- R is already expanded words

Algorithm 5 MMRE Algorithm

```
Input: Dictionary $D$, result set size $k$
Output: Expanded words $R$, $|R| = k$
R ← InitializeEmptyResultList()
$i \leftarrow 0$
while $i < k$ do
    FIND $w^* \leftarrow argmax_{w \in D}\, f_{mmre}(w)$
    $D \leftarrow D - w^*$
    $R \leftarrow R\, U\, w^*$
    $i \leftarrow i + 1$
end while
return $R$
```

During our query expansion experiments, we used word embedding technique [22], [23]. In word embedding, words are represented as a multi dimensional vectors, which helps the task of similarity computation among words [24]. In this model of representation of words, similar words have similar vectors. We used GloVe dictionary to represent word embeddings of words in query sets, as GloVe is shown to outperform earlier word representation models [25].

CHAPTER 3

ANALYZING CSI PERFORMANCE WITH QUERY EXPANSIONS

In this chapter, we will present different CSI creation techniques that we used during our experiments. First we will explain how we created the query-biased access count based CSI and pagerank based CSI. We used these CSIs to compare the impact on overall diversification performance by running over different query sets. In addition to the MMRE query expansion algorithm, we also applied a query expansion algorithm based on only word similarities that will be explained in this chapter.

3.1 Query Biased - Access Count Based CSI

As explained in Chapter 2, we adapted random based CSI in our work. However, this approach mainly depends on random function, and it always generates different document set. To create a more robust and more representative CSI, we applied a query-biased technique and generated an access count based CSI, we showed this technique in Algorithm 6.

For this technique, first of all we needed a query set to collect return set. We used AOL query set, which includes approximately 100.000 queries. For each query, we counted how many times each document appeared in top 10 of query results.

After we retrieved the query-biased access counts of every documents in the total collection, we created two CSIs. First as we show in Algorithm 6, each cluster is represented same by applying sample rate in cluster level. In addition to this, as shown in Figure 3.1, we also collected best %1 of the documents in that list. In this work, we used the second alternative. We removed spam documents from the CSI.

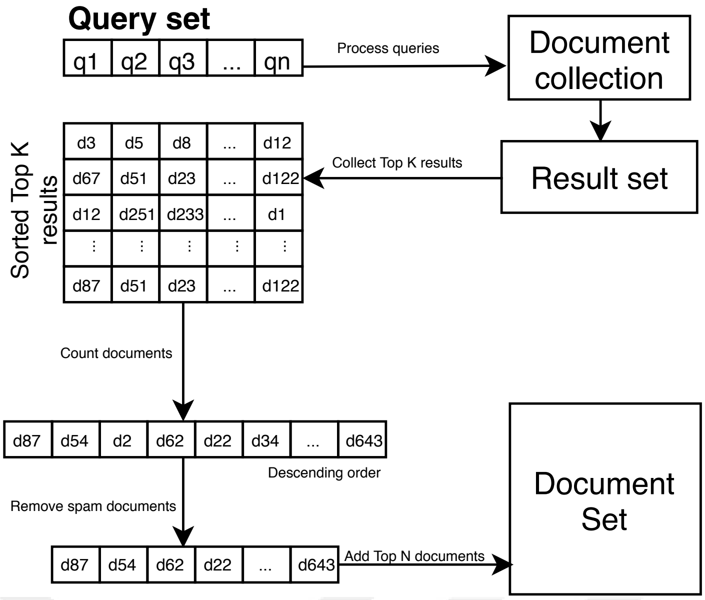


Figure 3.1: Query-biased Account count based CSI simulation.

Spam documents are retrieved from Cormack [26]'s studies which based on the same dataset.

Algorithm 6 Query-biased Access Count based CSI

Input: Query Set Q, Document Collection D, Number of Clusters K, Sampling Rate S

Output: Document Set DS

```
for Query $q \in Q$ do
    $Result_i \leftarrow$ ProcessQuery($D$, $q$)
end for
$POPULARDOCS \leftarrow$ COUNT in $Result_n$
for $k \in \{1, \dots, K\}$ do
    $SamplingCount_i \leftarrow S$ x $NumberofDocuments \in Shard_i$
    $DS \leftarrow$ TOP $SamplingCount_i$ $Documents \in Shard_i$
end for
return $DS$
```

3.2 Pagerank Based CSI

There is also another way to detect popular documents in the collection. We used pagerank information of documents in the collection. %1 of the all documents are collected from the collection according to their Pagerank scores. Highest scored documents are selected first. We applied spam prunning as we mentioned in the case of the access-based CSI.

3.3 Query Expansion with Word Embeddings

As we mentioned in Chapter 2, we adapted MMRE algorithm and applied it using word embeddings to generate different query sets. In addition to this diversified query set, we also generated another set which based on similarities between query and dictionary words.

For each query, first we simplified them by taking the average of word vectors. So that, each query is represented as a single vector. Then for each word in the dictionary, we computed similarity scores using word embeddings. Each query is expanded by an extra 5-words which are closest to the query vector.

Here, together with CSI techniques, we wanted to investigate the effect of expanded query sets on overall diversification metrics. It's important to compare them together since we processed these expanded query sets over CSI for selective search. We aimed to reach more precise clusters by processing the expansions over CSI, since expansions handle possible vocabulary gaps.

CHAPTER 4

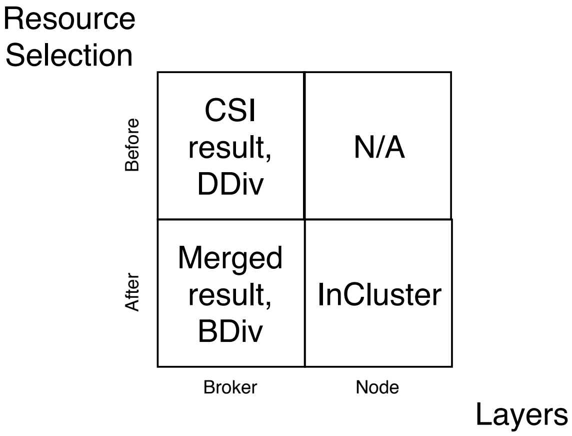
SEARCH RESULT DIVERSIFICATION FOR SELECTIVE SEARCH

In this chapter, we will explain different diversification approaches that can be applied on different layers of selective search. In addition to traditional approach, we adapted resource level diversification technique from [20]. Moreover, we also applied in-cluster level diversification. In the first section, we will discuss selective search along with search result diversification. Then, we will introduce three different diversification approaches that reveal the effectiveness of diversification at different stages of selective search.

4.1 Search Result Diversification for Selective Search

Traditionally, textual search systems divides collections into subsets, and the query is redirected through these collections. Even though, there is a gain in parallel query processing for this approach, it is still an exhaustive method. The query is processed for all documents. On the other hand, this computational cost can be redacted by only searching relevant shards among all. This approach, namely selective search, as introduced by Kulkarni and Callan [1], partitions collections into different clusters. Each cluster is represented by sampled documents in the centralized sampled index (CSI). Query is managed by broker. Broker processes the query over CSI, and selects best clusters. In Figure 2.1 it is shown after query is processed over CSI, only selected clusters will be chosen as a target.

Search result diversification's primary aim is always to serve most optimum results with respect to diversity and relevance metrics. To achieve this, for query result lists, recent diversification algorithms mainly re-rank them. Many prior work determine



Resource Selection	Broker	Node
Before	CSI result, DDiv	N/A
After	Merged result, BDiv	InCluster

Figure 4.1: Search result diversification at different stages and layers.

the best document using relevance and diversity estimation altogether in the linear equation [17]. Existing search result diversification algorithms are splitted into two main categories: implicit and explicit. In this work, we applied both of them.

It is possible to apply these search result diversification algorithms in different query result set. Naini et al. [27] studied the performance of diversification for exhaustive search by applying diversification at broker and nodes. For selective search; as we show in Figure 4.1; there exist different candidate sets for diversification. Broker produces result set by processing the query over CSI; moreover it also merges the result sets from chosen clusters. So that, at broker layer, CSI result diversification and merged result diversification are possible.

Furthermore, queries are processed inside the clusters, therefore these results can also be diversified before sending them back to broker. It is shown in Figure 4.1 as InCluster.

4.2 Diversification at Broker Node

First, we applied straightforward approach for search result diversification for selective search. This approach diversifies the final result set at broker node, just after collecting the results from selected clusters.

As shown in Figure 4.2, query is first processed over CSI. Resource selection algo-

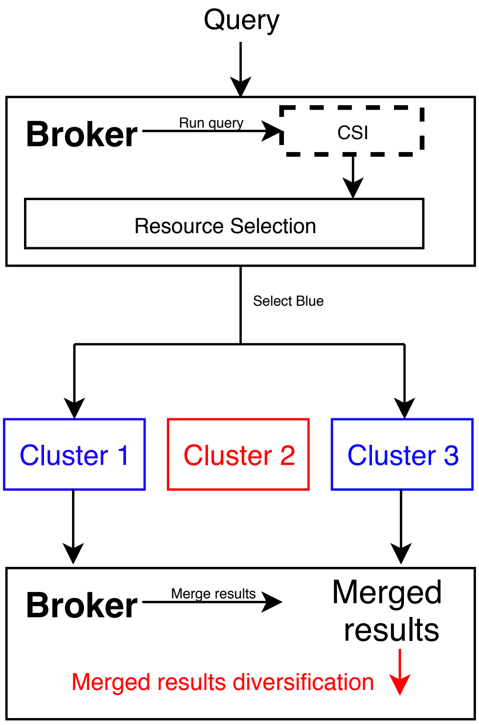


Figure 4.2: Diversification of merged results at Broker node.

rithm chooses best clusters using this result set. Then broker forwards the query to the selected clusters. Each cluster runs the query over their document collection, then returns their result set back to the broker. Broker merges all these results from different clusters. Then, diversification is applied over this merged result set at broker node.

4.3 DDiv: Diversification Based on CSI

As we discussed in Chapter 2, Hong and Si [20] proposed a different diversification approach for selective search. In their work, they showed DDiv can outperform traditional methods and their other proposed method, DivS as well. Therefore, we adapted DDiv which applies diversification before the relevant clusters are selected.

As shown in Figure 4.3, query is processed over CSI. Then, diversification is applied directly over this result set. Clusters will be chosen using this diverse set by resource selection algorithms. This approach diversifies in resource level to find diverse clusters and subsequently, final rankings as well.

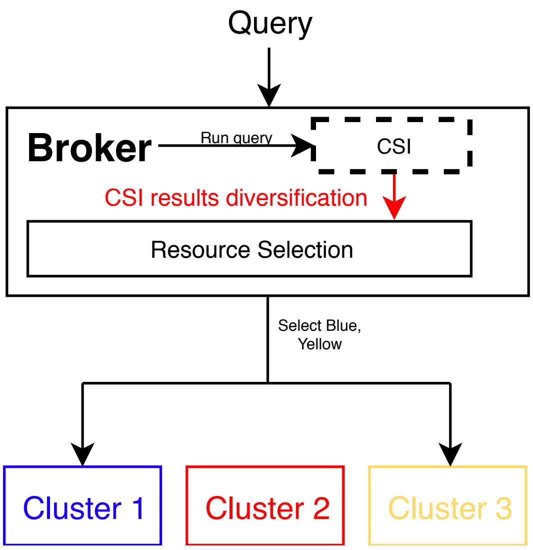


Figure 4.3: DDiv: Diversification based on CSI results.

4.4 Diversification inside the Selected Clusters

We applied another result level diversification approach in our experiments. Unlike the first straightforward approach, this method diversifies the result sets inside each clusters.

As shown in Figure 4.4, after query is forwarded to the relevant clusters by broker, each cluster diversifies its own result set for this query. Then, they return diversified set to the broker. This method can explore more distant documents, since each of clusters are represented with their diversified set at broker node.

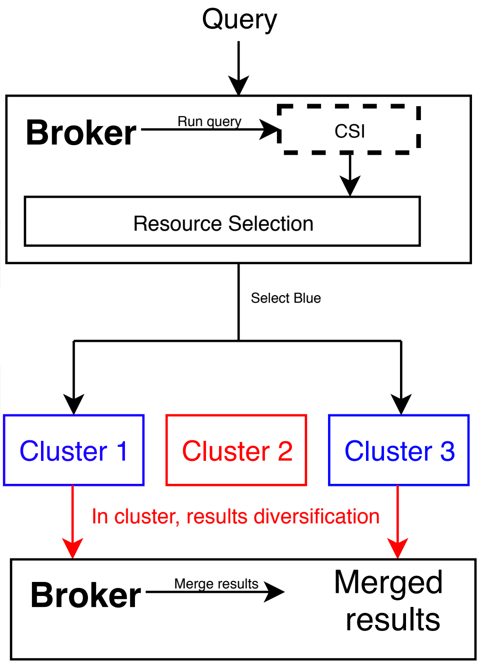


Figure 4.4: Search result diversification inside clusters.

CHAPTER 5

EXPERIMENTAL SETUP

In this chapter, we will explain our all experimental setup. First, we will mention about data set we used. Then respectively, clustering technique, different CSIs, query expansion, query processing, resource selection and diversification approaches will be explained. Finally, in the last section we will describe evaluation metrics.

5.1 Data Set

During the experiments, we used Clueweb B as a document collection[1]. There are total of 50,220,538 documents exist in this collection. Total number of terms for this collection is 163,629,158.

We used spam list, generated by [26] to prune spam documents in our experiments. We also adapted pagerank scores of Clueweb B documents that is available online[2].

We used Trec Web Track query sets for query processing [28, 29, 30, 31]. Trec query set 2009, 2010, 2011, 2012 are combined for experiments. They are also available online[3]. These 4 sets have total of 198 queries. For our explicit diversification experiments, we also used query aspects of this query sets.

Moreover, we also used AOL query set to create a query biased CSI [32]. AOL query set includes 100000 queries. We used a different query set, otherwise CSI could be biased over same query set. This makes evaluation difficult, especially for baseline comparisons, because query processing over that CSI would already return

[1] https://lemurproject.org/clueweb09.php
[2] https://lemurproject.org/clueweb09/pageRank.php
[3] https://trec.nist.gov/data/webmain.html

best popular documents.

5.2 Document Allocation

To simulate complete selective search framework, we created topic based clusters from Clueweb B document collection. We used K-means algorithm as we discussed in Chapter 2. We used a subset of documents from the collection to apply K-means. These documents are selected randomly. Moreover, each cluster centroid is also initialized randomly from this sample set. Then, rest of the sample set is also distributed to the clusters. Here, we used Kullback-Liebler divergence as described by Kulkarni and Callan [5] while computing the similarities between documents and centroid of clusters. Then K-means applied multiple times.

In his work, Hafızoğlu [33] proved that under selective search environment, best precision and diversity scores are achieved by splitting this dataset into 100 clusters. We used very same clusters with his work.

As we mentioned in Chapter 3, we applied spam prunning all three indexes.

5.3 Centralized Sample Indexes

After documents are allocated according to their topics, we created 3 different centralized sample indexes. First CSI is created by random sampling. Each cluster is represented in CSI by 1% sampling rate. As a result of this sampling, CSI includes 502,200 documents.

In addition to random CSI, we also created a CSI which is created using AOL query set. This set is processed over all documents in the collection and according to their number of existence in the top 10, best documents are selected from each cluster. Similarly, we applied same strategy to create Pagerank CSI. However, this time we used pagerank information of Clueweb B dataset.

For search result diversification method comparisons, we used random based CSI. At the end of the Chapter 6, we also showed some additional comparisons between access and random based CSIs.

5.4 Query Expansions using Word Embeddings

As we explained in Chapter 3, we used Global Vectors for Word Representations, GloVe[4] on our query expansion experiments. We used 6B tokens, that includes 400,000 words in the dictionary. In this representation, each entity is represented by 100 dimensional vectors.

We created 2 new query sets that derived from Trec query set. For both, we used cosine similarity to compute relevancy between words and the query average. For diversified expansion set, we diversified the most relevant 50 words to the query average vector. For MMRE function, we set lambda as 0.5.

5.5 Query Processing

As a query processing algorithm, Best Matching 25 is used [34]. Constants $k1$ and b is set to 1.2 and 0.5, respectively. We cleaned stop words from texts during query processing.

5.6 Resource Selection

We used Redde as a resource selection algorithm, since prior research favors it [20]. Hafızoğlu [33] made a cluster coverage analyze in same dataset, which shows that top $\%10$ of these clusters have the $\%99$ of relevant documents. That's why we also picked the best $\%10$ of clusters. Each cluster returns their best 20 results, that's why our evaluations are mostly based on top 20, 10 and 5 results. We applied Redde algorithm for top 200 documents of the result set.

[4] https://nlp.stanford.edu/projects/glove/

5.7 Diversification

We applied both implicit and explicit search result diversification algorithms in this study. As an implicit algorithm, SY is implemented. xQuAD algorithm is also implemented, which is a very successful example of explicit algorithms. These algorithms are used as explained in Chapter 2. For xQuAD, BM25 query processing scores are normalized by sum of scores in result set. For both algorithms, we used λ values: 0.25, 0.5 and 0.75. Except the adapted DDiv method, we always diversified top 100 documents of the result set. For DDiv method, since diversification is applied directly over CSI, we diversified top 200 of CSI result set.

5.8 Evaluation Techniques

We used α-nDCG as a diversity performance metric for all experiments [35]. Using this technique, we showed diversity scores for top 5, 10 and 20 results. We used nDeval[5] as a tool from Trec Web Track archives. nDeval calculates diversity metrics for a given query results, including α-nDCG.

In addition to diversity metrics, we also showed precision values for top 5, 10 and 20 results. These scores are computed using a tool named, trec_eval, which is also part of Trec Web Track. For both evaluation metrics, Trec provides a ground truths to compare.

[5] http://www-personal.umich.edu/ kevynct/trec-web-2014/

CHAPTER 6

EXPERIMENTAL RESULTS

In this chapter, we will present our experimental results. First, we will present results for different CSIs together with query expansions. Next, we will compare different search result diversification methods.

6.1 CSI Results

We created random, access and pagerank based CSIs. We compared them under selective search environment as described in Chapter 2. In tables, **mmre** represents the diversified query expansions. Other expanded set we described in Chapter 3, is named as **exp**.

According to their effectiveness values in Table 6.1, access based beats other CSI alternatives.

We also analyzed CSI performances with query expansions. In Tables 6.2 and 6.3, it is shown that access based CSI can not gain from query expansions. On the other hand random and pagerank CSIs can improve overall diversity metrics for selective

Table 6.1: Effectiveness Results for CSI types.

	P@5	P@10	P@20	a-nDCG@5	a-nDCG@10	a-nDCG@20
access	**0.3381**	**0.3244**	**0.2865**	**0.2811**	**0.3162**	**0.3489**
random	0.3289	0.3162	0.2825	0.2758	0.3129	0.3438
pagerank	0.3188	0.3025	0.2617	0.2639	0.3013	0.3295

Table 6.2: Diversity effectiveness results for CSI types with query expansions.

csi	query expansion	a-nDCG@5	a-nDCG@10	a-nDCG@20
access	-	**0.2811**	**0.3162**	**0.3489**
	mmre	0.2791	0.3132	0.3439
	exp	0.2796	0.3133	0.3404
pagerank	-	0.2639	0.3013	0.3295
	mmre	0.2742	0.3117	**0.3392**
	exp	**0.2782**	**0.3121**	0.3388
random	-	0.2758	0.3129	0.3438
	mmre	0.2816	**0.3171**	**0.3481**
	exp	**0.2819**	0.314571	0.3440

Table 6.3: Precision effectiveness results for CSI types with query expansions.

csi	query expansion	P@5	P@10	P@20
access	-	**0.3381**	**0.3244**	**0.2865**
	mmre	0.3239	0.3112	0.2764
	exp	0.3289	0.3076	0.2685
pagerank	-	0.3188	0.3025	0.2617
	mmre	0.3218	**0.3091**	**0.2665**
	exp	**0.3259**	0.3025	0.2579
random	-	0.3289	0.3162	**0.2825**
	mmre	0.331	0.3173	0.2812
	exp	**0.3401**	**0.3193**	0.2789

search.

In conclusion, we found that access based CSI can outperform other CSI techniques both in diversity and precision metrics. However, we also realized that it is possible to improve the overall diversity quality by expanding query words using word embeddings. Random based CSI with mmre query expansions outperform all other alternatives at top 10 and top 5 results and it is barely beaten by access CSI at top 20 results. Therefore, random based CSI is a better choice in case of query expansions will run over CSI.

Table 6.4: Diversity effectiveness results for Implicit Diversification.

	lambda	a-nDCG@5	a-nDCG@10	a-nDCG@20
BDiv	0.25	0.1899	0.1892	0.1924
	0.5	0.2338	0.2525	0.2685
	0.75	**0.2665**	**0.3048**	**0.3286**
DDiv	0.25	0.2787	**0.3145**	**0.3453**
	0.5	**0.2801**	0.3126	0.3452
	0.75	0.2754	0.3097	0.3427
InCluster	0.25	0.2576	0.27463	0.2808
	0.5	0.2626	0.2908	0.3106
	0.75	**0.2705**	**0.3107**	**0.3397**

6.2 Implicit Diversification Results

We compared methods described in Chapter 4 in the implicit setup. We used these abbrebiations to identify methods in the tables: **BDiv** for *Diversification at Broker Node*; **DDiv** for Hong and Si [20]'s *Diversification approach based on sample Documents*; **InCluster** for *Diversification inside the Selected Clusters*.

In the Tables 6.4 and 6.5, it is shown that DDiv method beats their compatitors as Hong and Si [20] claimed in their work. After we verified this, we compared *mmre* method with them. This showed us that adapted method **mmre** can outperform best scores of all these methods, as shown in Table 6.6.

Since *mmre* beat other methods, we combined this resource level diversification method with *BDiv* method. Here, we aimed to outperform *mmre* performance by applying a diversification to the final result set at broker node. This method represented as **BDiv+mmre** in the tables. Also, other resource level diversification method, *DDiv* could also improve its result set in same way. Therefore, we applied same strategy this method as well. In the tables, its abbreviation is **DDiv+BDiv**. None of these additional methods could beat mmre approach in implicit setup.

Table 6.5: Precision effectiveness results for Implicit Diversification.

	lambda	P@5	P@10	P@20
BDiv	0.25	0.1279	0.0741	0.0411
	0.5	0.2081	0.1645	0.1112
	0.75	**0.2792**	**0.2401**	**0.1812**
DDiv	0.25	0.3279	0.3122	0.2782
	0.5	**0.3289**	**0.3137**	0.2812
	0.75	0.3259	0.3132	**0.2815**
InCluster	0.25	0.2426	0.1665	0.0959
	0.5	0.2538	0.2091	0.1475
	0.75	**0.2904**	**0.2553**	**0.2003**

Table 6.6: Effectiveness results for Implicit Diversification with all methods' best results.

	P@5	P@10	P@20	a-nDCG@5	a-nDCG@10	a-nDCG@20
BDiv	0.2792	0.2401	0.1812	0.2665	0.3048	0.3286
DDiv	0.3279	0.3122	0.2782	0.2787	0.3145	0.3453
InCluster	0.2904	0.2553	0.2003	0.2705	0.3107	0.3397
mmre	**0.331**	**0.3173**	**0.2812**	**0.2816**	**0.3171**	**0.3481**
BDiv+mmre	0.2822	0.2371	0.1799	0.2745	0.3078	0.3296
DDiv+BDiv	0.2731	0.2365	0.1802	0.2653	0.3029	0.3283

Table 6.7: Diversity effectiveness results for Explicit Diversification.

	lambda	a-nDCG@5	a-nDCG@10	a-nDCG@20
BDiv	0.25	0.3001	0.3368	0.3738
	0.5	0.3163	0.3514	**0.3826**
	0.75	**0.3215**	**0.3522**	0.3824
DDiv	0.25	0.2655	0.2998	0.3320
	0.5	0.2695	0.3047	0.3354
	0.75	**0.2729**	**0.3071**	**0.3392**
InCluster	0.25	0.2842	0.3197	0.3552
	0.5	0.2918	0.3268	0.3613
	0.75	**0.2945**	**0.3308**	**0.3647**

6.3 Explicit Diversification Results

In the explicit setup, we found that BDiv outperforms other methods. In Tables 6.7 and 6.8, it is shown that DDiv method is actually behind the others with respect to precision and diversity.

As we have shown in previous section, we merged some methods together to improve their performances. In explicit setup, BDiv method clearly beats others, therefore we used *mmre* method together with BDiv to improve cluster selection. BDiv results could also improve by combining it with DDiv. Since it can also improve cluster selections. That's why, we investigated their results, too. In the Table 6.9, it is shown that none of the methods could actually beat BDiv method.

As an additional experiment, we employed access based CSI for the best methods found above. In Table 6.10, we showed their results in implicit setup. Best scored method, DDiv with random based CSI still outperforms access based CSI. For explicit setup, similarly, BDiv with random based CSI beats its access based alternative as shown in Table 6.11.

Table 6.8: Precision effectiveness results for Explicit Diversification.

	lambda	P@5	P@10	P@20
BDiv	0.25	0.3675	0.3523	**0.3117**
	0.5	0.3766	**0.3533**	0.3033
	0.75	**0.3838**	**0.3533**	0.2957
DDiv	0.25	0.3157	0.3076	0.2734
	0.5	0.3198	0.3102	0.2744
	0.75	**0.3279**	**0.3112**	**0.2749**
InCluster	0.25	0.3371	0.3223	0.2845
	0.5	0.3431	0.3218	0.2853
	0.75	**0.3472**	**0.3269**	**0.2873**

Table 6.9: Effectiveness results for Explicit Diversification with all methods' best results.

	P@5	P@10	P@20	a-nDCG@5	a-nDCG@10	a-nDCG@20
BDiv	**0.3766**	**0.3533**	**0.3033**	**0.3163**	**0.3514**	**0.3826**
DDiv	0.3279	0.3112	0.2749	0.2729	0.3071	0.3392
InCluster	0.3472	0.3269	0.2873	0.2945	0.3308	0.3647
mmre	0.331	0.3173	0.2812	0.2816	0.3171	0.3481
BDiv+mmre	0.3665	0.3472	0.2967	0.3154	0.3475	0.3789
DDiv+BDiv	0.3706	0.3447	0.2954	0.3088	0.3463	0.3786

Table 6.10: Effectiveness results for Access and Random based CSIs for Implicit Diversification.

	csi	P@5	P@10	P@20	a-nDCG@5	a-nDCG@10	a-nDCG@20
BDiv	access	**0.2832**	0.233	0.1805	**0.2705**	0.3036	**0.3287**
	random	0.2792	**0.2401**	**0.1812**	0.2665	**0.3048**	0.3286
DDiv	access	**0.331**	0.3112	**0.281**	0.2773	0.3130	0.3447
	random	0.3279	**0.3122**	0.2782	**0.2787**	**0.3145**	**0.3453**

Table 6.11: Effectiveness results for Access and Random based CSIs for Explicit Diversification.

	csi	P@5	P@10	P@20	a-nDCG@5	a-nDCG@10	a-nDCG@20
BDiv	access	0.3645	0.3503	0.297	**0.3166**	**0.3525**	0.3799
	random	**0.3766**	**0.3533**	**0.3033**	0.3163	0.3514	**0.3826**
DDiv	access	**0.335**	**0.3137**	**0.2789**	**0.2769**	**0.3104**	**0.3422**
	random	0.3279	0.3112	0.2749	0.2729	0.3071	0.3392

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis, we provided an in-depth analysis of search result diversification in the context of selective search, and proposed extensions to improve diversification effectiveness. First, we showed that creating a CSI based on the past document access statistics yields both more relevant and diverse results than a CSI based on randomly sampled documents. However, by processing queries expanded with diverse terms during resource selection, it is also possible to obtain diversification performance that is comparable to the latter. This finding is also important to show that even when such past statistics are not available, a random CSI together with diversified query expansion performs reasonably well.

Second, we investigated the diversification performance at different layers (i.e., at the broker vs. in the clusters) and at different stages, namely, before resource selection and before/after result merging. We found that when a representative implicit diversification method, namely, SY, is employed; the best diversification performance is obtained by selecting diverse resources as suggested by Hong and Si [20]. While doing so, simply processing expanded queries with diverse terms over the CSI, as we proposed in this thesis, outperform the previous approach in Hong and Si [20]. Interestingly, the findings vary for the explicit diversification. By employing a representative explicit approach, xQuAD, we demonstrated that diversifying merged results at the broker is superior to diversifying partial results at the clusters or diversification during the resource selection. This is again a new and contradicting finding with respect to Hong and Si [20], and implies that when there are more clues for diversification, it is better to conduct it at a more fine-grain level, i.e., over the result list, rather than attempting to diversify resources as a whole.

There are various future research directions for our work. In particular, in our additional experiments we observed that by conducting a selective expansion of queries, i.e., expanding only a subset of them and/or setting different thresholds for expansion terms on a per query basis, it is possible to further improve the diversification effectiveness. We plan to explore such selective expansion approaches as our future work. As a second research direction, we will investigate the diversification efficiency for selective search. For instance, it is possible to create summary vectors for the documents to conduct the diversification methods at the broker more efficiently. Such optimizations are also left for our future work.

REFERENCES

[1] Anagha Kulkarni and Jamie Callan. Selective search: Efficient and effective search of large textual collections. *ACM Transactions on Information Systems (TOIS)*, 33(4):17, 2015.

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

[3] Arbi Bouchoucha, Jing He, and Jian-Yun Nie. Diversified query expansion using conceptnet. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1861–1864. ACM, 2013.

[4] Xiaoyong Liu and W Bruce Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193. ACM, 2004.

[5] Anagha Kulkarni and Jamie Callan. Document allocation policies for selective searching of distributed indexes. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 449–458. ACM, 2010.

[6] Jinxi Xu and W Bruce Croft. Cluster-based language models for distributed retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 254–261. ACM, 1999.

[7] Robin Aly, Djoerd Hiemstra, and Thomas Demeester. Taily: shard selection using the tail of score distributions. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 673–682. ACM, 2013.

[8] Milad Shokouhi, Luo Si, et al. Federated search. *Foundations and Trends® in Information Retrieval*, 5(1):1–102, 2011.

[9] James P Callan, Zhihong Lu, and W Bruce Croft. Searching distributed collections with inference networks. In *ACM SIGIR Forum*, volume 51, pages 160–167. ACM, 2017.

[10] Luo Si and Jamie Callan. Relevant document distribution estimation method for resource selection. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 298–305. ACM, 2003.

[11] Jaime Arguello, Jamie Callan, and Fernando Diaz. Classification-based resource selection. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1277–1286. ACM, 2009.

[12] Jaime Arguello, Fernando Diaz, Jamie Callan, and Jean-Francois Crespo. Sources of evidence for vertical selection. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322. ACM, 2009.

[13] Milad Shokouhi. Central-rank-based collection selection in uncooperative distributed information retrieval. In *European Conference on Information Retrieval*, pages 160–172. Springer, 2007.

[14] Jangwon Seo and W Bruce Croft. Blog site search using resource selection. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1053–1062. ACM, 2008.

[15] Dorit S Hochbaum. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.

[16] Rodrygo LT Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. Explicit search result diversification through sub-queries. In *European conference on information retrieval*, pages 87–99. Springer, 2010.

[17] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, pages 381–390. ACM, 2009.

[18] Ke Tao, Fabian Abel, Claudia Hauff, Geert-Jan Houben, and Ujwal Gadiraju. Groundhog day: near-duplicate detection on twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1273–1284. ACM, 2013.

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

[20] Dzung Hong and Luo Si. Search result diversification in resource selection for federated search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 613–622. ACM, 2013.

[21] Jaime G Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, volume 98, pages 335–336, 1998.

[22] Saar Kuzi, Anna Shtok, and Oren Kurland. Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1929–1932. ACM, 2016.

[23] Kezban Dilek Onal, Ismail Sengor Altingovde, and Pinar Karagoz. Utilizing word embeddings for result diversification in tweet search. In *AIRS*, pages 366–378. Springer, 2015.

[24] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, 2014.

[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://www.aclweb.org/anthology/D14-1162`.

[26] GV Cormack. Waterloo spam rankings for the clueweb09 dataset@ online, 2009.

[27] Kaweh Djafari Naini, Ismail Sengor Altingovde, and Wolf Siberski. Scalable and efficient web search result diversification. *ACM Transactions on the Web (TWEB)*, 10(3):15, 2016.

[28] Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the TREC 2009 web track. In *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA*, 2009.

[29] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Gordon V. Cormack. Overview of the TREC 2010 web track. In *Proceedings of The Nineteenth Text REtrieval Conference, TREC 2010, Gaithersburg, Maryland, USA*, 2010.

[30] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Ellen M. Voorhees. Overview of the TREC 2011 web track. In *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA*, 2011.

[31] Charles L. A. Clarke, Nick Craswell, and Ellen M. Voorhees. Overview of the TREC 2012 web track. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA*, 2012.

[32] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *InfoScale*, volume 152, page 1, 2006.

[33] Fatih Hafızoğlu. Improving the efficiency of distributed information retrieval using hybrid index partitioning. Master's thesis, 2018.

[34] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

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