PERSONALIZATION AND DIVERSIFICATION OF SEARCH RESULTS

by

Naveen Kumar

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Winter 2013

PERSONALIZATION AND DIVERSIFICATION OF SEARCH RESULTS

by

Naveen Kumar

Approved: _____
Ben Carterette, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Errol Lloyd, Ph.D.
Chair of the Department of Computer Science

Approved: _____
Babatunde Ogunnaike, Ph.D.
Interim Dean of the College of Engineering

Approved: _____
Charles G. Riordan, Ph.D.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

I would like to express my gratitude towards my advisor, Dr. Carterette, for his excellent guidance, patience, and providing me an opportunity to work in the field of information retrieval (IR). My real interest in IR developed in summer 2011 during TREC conference meetings. During work on TREC 2011, I got convinced to work on master thesis as IR is a fascinating area of research with lot of questions still unanswered, and with very interesting problems to study.

I would like to thank my committee members Dr. Mccoy and Dr. Fang for valuable inputs. I am thankful to Dr. Wisser for providing the opportunity in his lab and valuable discussions on different aspects of academic research.

I would like to thank my parents, brother and wife for their constant motivation, support whenever needed, as it is hard to be away from family.

I would like to thank all IR Lab members specially Praveen and Ashwani for helping with diversity aspect of retrieval.

I would like to thank Dr. David Vallet for providing the corpus with user and topical judgments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

There has been lot of research in the area of information retrieval on different aspects of search such as personalization, diversity, evaluation measures etc. In this thesis, we hypothesize that personalization and diversification can coincidently exist with each other. We propose two novel approaches, one for personalization by incorporating feedback from query logs of similar users by extending the state art of personalization method, other for subtopic retrieval using N-grams as document representatives for diversity.

There is a general consensus among researchers that personalization and diversity are opposed to each other since personalization advocates for information based on user interests while diversity support the maximum information gain for a given query by selecting documents which incorporate all perspectives of query. Our model aims to provide the users with maximum diverse information with consideration of user interests. For example, for a given a query "RSS" which has numerous meanings such as Rich Site Summary, Rashtriya Swayamsevak Sangh, Remote Sensing Service etc., the proposed system output should accommodate not only different aspects of the query in the output results, but also consider user interests. Given the above mentioned query, for users interested in politics, documents with the Rashtriya Swayamsevak Sangh aspect should be ranked higher compared to documents related to the technical perspective, i.e., Rich Site Summary.

Chapter 1

INTRODUCTION

1.1 Introduction to IR Systems

An Information Retrieval system can be defined as a system for finding relevant information from a large unstructured textual collection based on user needs. The term unstructured refers to documents which are not well-defined syntactically, have data with a given semantics. IR plays an important role for human beings in different aspects of search like web search on World Wide Web (WWW) and product search on commercial web sites such as Amazon, Best Buy, eBay etc. It is a multifaceted area of research with sub-areas such as personalization, diversity, index optimization, handling big data, machine learning approaches in retrieval and evaluation measures. The multifaceted fact can be measured from the fact that in the mentioned application areas, the underlying retrieval models and information needs are different. In web search, the users are searching from a collection created by crawling billions of web pages. The considerations of search models on the web are such things as document links with each other, queries being navigational, informational or transactional etc. while different factors are important in case of product search such as cost of product, reviews of product, novelty of product etc.

Information retrieval has been an area of research for scientists since 1951. In earlier stages, most of the information retrieval systems were based on library collections, scientific journals etc. There has been significant change in the approach for retrieving information since that time. With the advent of web search engines which allow users to search on the web, lots of companies have invested in this field. Some of the commercial search engines currently used include Google, Yahoo and Bing etc. Commercial term in the above statement doesn't mean that these systems charge money

from people to search on web, but they do make revenue by showing advertisements with the search results.

1.2 History

The term "Information Retrieval" was coined by Calvin Mooers in about 1950, when he was writing his master thesis at MIT. According to Mooers, "Information Retrieval is the name for the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him". In the earlier stages, information retrieval was restricted to academia search for papers, journals and other articles. The evolution of information systems started from manual systems; later on, when the computational capabilities increased, IR systems evolved into complex computation systems. With the advent of modern computationally powerful machines, the mentioned systems are able to deal with terabytes of data in a very short span of time. For example, TREC 1.5, Terabyte Track brought forth the challenge of handling Terabyte data, which demanded researchers to built frameworks for handling such large datasets. Hadoop is an example of such a framework that allows distributed processing of large data sets across clusters of computers using simple programming models. Other frameworks which are built on hadoop are Pig and Hive.

1.3 Retrieval Models in IR

Retrieval models can be categorized as boolean, vector space or probabilistic. In a boolean model, queries are formulated as a boolean combination of terms. For example, a query $(t_1 \; and \; t_2) \; or \; t_3$ can be satisfied by a document D_1 if and only if the document contains either all terms $t_1,\ t_2$ and t_3 or terms t_1 and t_2 or the term t_3. The main limitation with this approach is that documents cannot be ranked by relevance. In the vector space approach, each term of document is assigned a numerical weight estimating the term usefulness. The numerical weight of a term can be calculated in two ways as term frequency (tf) in a document or term frequency-inverse document

frequency ($tf - idf$) in whole corpus. The tf factor calculates term weight locally for a document, while idf accounts for the term representation in whole corpus. So, each document is represented as a vector containing weight of terms. The score between a document D_1 and query q is calculated by measuring the similarity between query vector and document vector. This similarity measure is also called as cosine similarity measure. The probabilistic based approach makes use of formal probability theory and statistics to arrive at the estimates of probability of relevance for ranking the documents. In a probabilistic method (language model), for a given document D and query q, the conditional probability $P(D|q)$ estimates the relevance of D to a given query q. Indri, an open source retrieval tool implemented in C is based on probability based methods of retrieval, while Lucene, another open source retrieval tool implemented in java is based on vector space retrieval model.

1.4 Recent Trends

Now a days, with lots of options at hand for end users, search engines attempt to incorporate many useful factors such as users query logs, click information from users, incorporating diversity information etc. Not only do they employ user information but they also track world events. For instance, the query "US Open" in the month of September could possibly mean tennis while the same query at a different time could suggest golf. Also, with the growth in the area of machine learning, researchers[26] have suggested ranking the documents by learning from different parameters such as term frequency, scores from different retrieval models etc.

1.5 Text Retrieval Conference

TREC (Text REtrieval Conference) is a conference co-sponsored by the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense started in 1992. As per NIST, its main goal is to enable researchers by providing infrastructure and different retrieval tasks such as diversity, twitter search, temporal summarization etc. Also, it helps to increase communication among industry,

academia, and government by creating an open forum for the exchange of research ideas.

Chapter 2

SYSTEM ARCHITECTURE

In this chapter, we discuss the architecture of an information retrieval system and its components.

2.1 Introduction

Architecture of an elementary information retrieval system as shown in Figure 2.1, may consist of four components. Details of each component are as below:

- **Corpus**: Corpus is a collection of documents. A document contains information in a format like XML, HTML etc. The corpus size may vary from a few documents to billions of documents (in web search). Each document in the corpus has a unique identification number which is returned by the retrieval model.

- **Index**: Queries contain terms. So, in order to get the document list which contain the query terms, we need indexes. Index represents the mapping of a term t to documents (which contains term t). It is created with the help of Indexer (a program written in any programming language). Index can be stored either in memory (called in memory indexes) or on disk (called disk indexes).

-

- Table 2.1 describes an elementary index. For example, term "algorithm" exists in WTX104B011368, WTX104B012134, WTX104B01497 and WTX104B01994.

- **Retrieval Model**: This is the most important component of a retrieval system. This calculates the score of each document for a given query and produces the

Term	Documents
algorithm	WTX104B011368, WTX104B012134, WTX104B01497, WTX104B01994
facbook	WTX104B011192, WTX104B011193, WTX104B011232, WTX104B012012, WTX104B01439
google	WTX104B01110, WTX104B01438
linkedin	WTX104B01110, WTX104B01438
news	WTX104B011031, WTX104B011201, WTX104B011202, WTX104B012134, WTX104B013957
music	WTX104B011202
florida	WTX104B01110, WTX104B011202
...	...

Table 2.1: Index Table

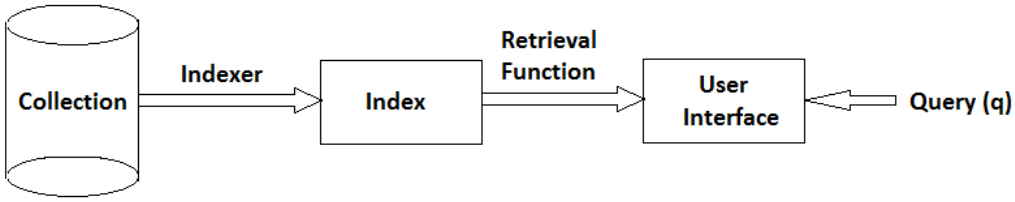


Figure 2.1: Basic IR system architecture

final rankings sorted by scores. Some examples of retrieval models are BM25, Language Model, MRF model etc.

- **User Interface**: User interface plays an important role in commercial search engines but generally is not required for research purposes.

2.2 Our Architecture

In this section, we discuss about the architecture of our system shown in figure 2.3 with all components. The system mainly consists of following components:

- **Collection**: Our corpus contains documents (in HTML format) crawled using Apache Nutch crawler [1]. More information about the collection can be found in section 6.1.1.

[1] Apache Nutch Crawler (http://apache.nutch.org)

- **Index**: From the corpus, we built ivory index using ivory tool. Ivory [27] is an open source information retrieval tool designed by Dr. Jimmy Lin group at University of Maryland, College Park.

- **Sparse Format**: We build sparse format using Apache Mahout API [2] since sparse format helps in getting more statistical information about the corpus conveniently compare to other formats like indexes. The main advantage is one time calculation of various parameters like document freq of term, tf-idf of term, term freq in a document etc. There are various output files in the sparse format like dictionary file, tokenized document files which are very helpful for building a statistical model. We used the Mahout API on top of Apache Hadoop.

- **MRF FD Model**:We used the MRF FD model suggested by Metzler et al. [5] for our base retrieval. MRF stands for Markov Random Field. FD stands for full dependence. MRF is a graph based retrieval model where random field is constructed by the graph nodes (random variables) containing query terms and document for which the score has to be estimated.

- **Pesonalization**: We used an extended state of art personalization based approach using clusters which is described in chapter 3.5.

- **Users Query Logs**: This module contains query logs from AOL search engine which are processed to make groups of different users where users in a group exhibit similar interests.

- **Sub Topic Retrieval**: This module retrieves the subtopics containing different aspects of query q. As shown in figure, 2.2 for query "apple", two subtopics can be apple fruit and apple company.

- **Diversity**: Diversity helps retrieval system for ranking results in such a manner that documents incorporate all subtopics (taking account the importance of a

[2] Apache Mahout (http://mahout.apache.org)

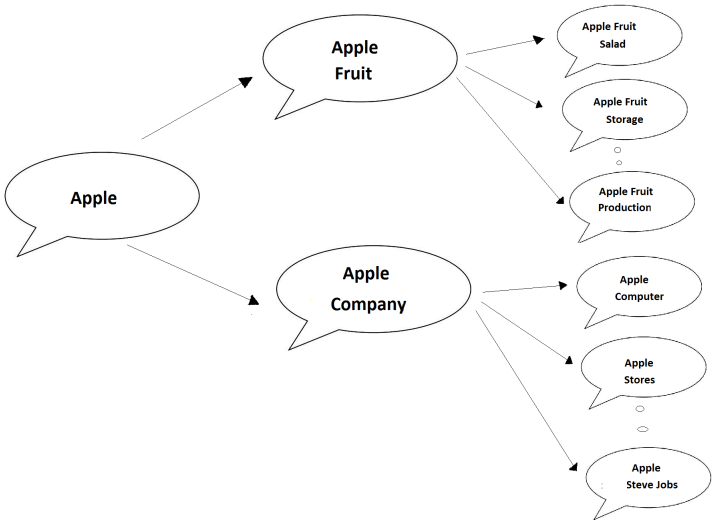


Figure 2.2: Diversity Example

subtopic) of query q. Not all diversity frameworks need subtopic retrieval. Our diversity module consists of xQUAD framework [21] which is discussed later in diversity chapter.

- **Final Ranking**: Both aspects of personalization and diversity are important for users. So, both aspects are combined linearly to get the final ranking. The effect of personalization on diversity and vice verse is studied in results section. 6.3.1.

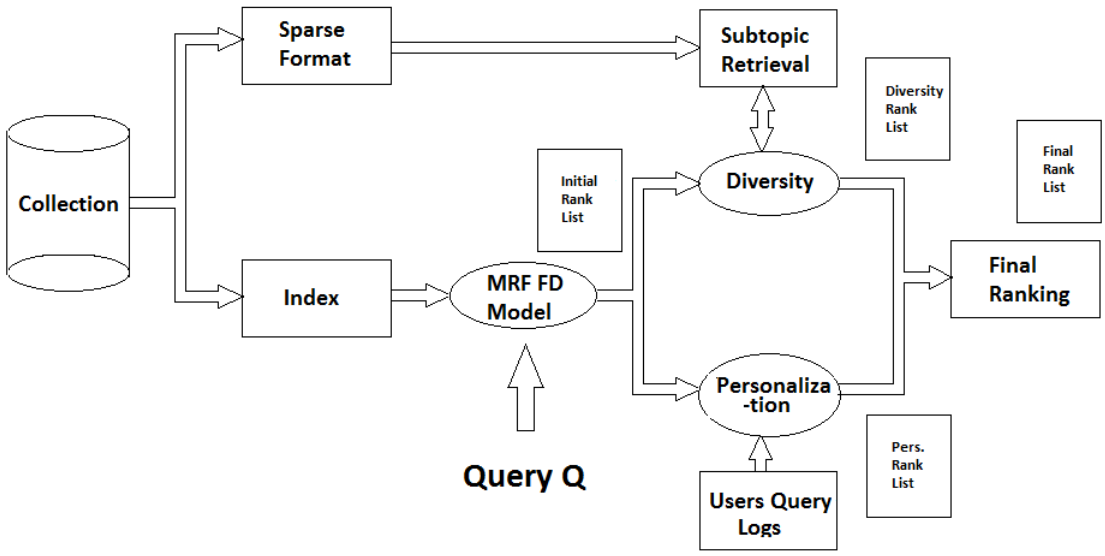


Figure 2.3: Our Retrieval System Architecture

Chapter 3

PERSONALIZED SEARCH

In this chapter, we discuss about the methods adopted by us for personalized search results.

3.1 Introduction

Personalized search is an important constituent of a search engine since it helps in improving results based on the user's recorded interests. Not only past search queries, but past clicked results can be used as implicit behavior for improvement of results by boosting the clicked results. The Potential of Personalization in a search engine has been studied by [3] where the authors suggest that the explicit judgments for a given query vary among different users. Another interesting aspect of personalization is the use of query expansion by mining user logs as suggested by Hang Cui [4] et al.. Implicit and explicit feedback from user behavior are important factors for personalization. There are some approaches which suggest that results can be improved using user modeling. There are different ways using which user models can be created on the basis of evidences such as Content, Behavior, and Context etc. Content based models exploit the user query history, behavior based models make use of clicked web pages while context models consider the location, time etc. factors. There are various other factors which account for personalization like re-finding which is very common on the web. As suggested by Teevan [3] et al., 33 percent of the queries on the web are repeated and 39 percent of the clicks are repeated clicks.

3.2 User Representation

Our corpus is comprised of two kinds of user sets as described in section 6.1.1. For one set of users, we have recent bookmarked history of last 15 days while for the other set we have query logs of each user for specific duration.

For set with query logs, a user is represented by concatenation of query logs. Suppose, a user has searched for "apartments in Newark" and "university of Delaware". The user is represented as "apartments in Newark university of Delaware".

For user set containing bookmarks, we have tags associated to each bookmarked URL. Bookmarked URLs have been tagged using delicious website, a social tagging system. User is represented by concatenation of tags. For example, suppose the user booked marked $http://thesaurus.com/browse/$ with tags "thesaurus dictionary online dictionary" and $http://www.bbc.co.uk/$ with tags "news UK online news'. The user is represented as "thesaurus dictionary online dictionary news UK online news". So, a user can bookmark multiple URLs and a URL may contain multiple tags whereas same tag can be contained multiple times in an URL.

3.3 Clustering Users

For grouping into similar groups, we cluster the represented users using k-means clustering algorithm Tapas et al.[24]. k-means clustering algorithm is provided with a set of users (u_1, u_2, ..., u_n), where each user is represented as a d-dimensional vector (containing terms). The algorithm iteratively partitions the n users into k sets (k $\leq$ n) S = S_1, S_2, ..., S_k such that the sum of squares within a cluster is minimum. The equation for k-means clustering algorithm can be given as:

$$argmin \sum_{i=1..k} \sum_{u_j \in S_i} ||u_j - \mu_i||^2 \tag{3.1}$$

In the equation 3.1, μ_i is the centroid of users in set S_i. Although there are various approaches which has been suggested for initial selection of points for better clustering results, our initial selection of k users was random. After the initial sets, each cluster centroid is calculated, and users are further grouped in different clusters

as centroids change based on users being added or removed from each cluster with iterations. In equation 3.1, we used cosine similarity measure for distance calculation between two users.

3.3.1 Distance Calculation Between Two Users

Let U = u_1, ..., u_n be a set of users and T = t_1, ...,t_m the set of distinct terms occurring in U. A user u is represented by m dimensional vector $\vec{tw}_u$. $\vec{tw}_u = (tw(t_1, u), ..., tw(t_m, u))$. Distance between two users is given by equation below:

$$Sim(\vec{u}_1, \vec{u}_2) = \frac{\vec{u_1} \cdot \vec{u}_2}{|\vec{u}_1| \times |\vec{u}_2|} \tag{3.2}$$

Equation 3.2 represents the similarity equation (cosine similarity).

$$tw(t_i, u_p) = tf(t_i, u_p) \times log(\frac{N}{df(t_i)}) \tag{3.3}$$

In equation 3.3 tw (t_i, u_p) defines the term weight of term i for user u_p. N is the total number of users in the dataset; df(t_i) is user frequency for term i.

3.4 Folksonomy Based Approach

Tagging has been an important feature of web 2.0. Del.icio.us is a social based tagging system where each user adds bookmarks and tags with keywords. For example, a user might bookmark "http://bbc.co.uk". Some of the tags which user might use are *news*, *english*, *UK*, *online*. Folkosonomy is a system of annotating and categorizing text using annotation of tags. This helps in understanding the user interests which can be incorporated in search. Tags act as metadata of a tagged document. $P(d|u)$ can be calculated using this approach by calculating the tag similarity between the document d and user u (since user representation also contain tags).

3.5 Proposed Model

In our personalized search, we extend the state art of personalization method. Given a query q and user u the state art of personalization method for a given document d as suggested by [1] can be defined as

$$p(d|q,u) = \frac{p(q|d,u)p(d|u)}{p(q|u))} \propto \frac{p(q|d)p(d|u)}{\sum_{d'} p(q|d')p(d'|u)} = \frac{p(d|q)p(d|u)}{\sum_{d'} p(d'|q)p(d'|u)} = C_1 p(d|q)p(d|u) \tag{3.4}$$

Equation 3.4 takes into account the query history of single user for a given query q. Our proposed approach makes use of query logs of similar users based on the distance among the users.

$$P_{pers} = \lambda P_{userpers} + (1-\lambda) P_{simuserspers} \tag{3.5}$$

In equation 3.5, λ is a variable, when set to 1 incorporates the query logs of user with query q only, when set to 0 incorporates the query logs of all similar users, except user u.

$P_{userpers}$ user personalization factor for user u can be given by equation 3.4 for a given quey q and document d. In order to calculate the other factor i.e. $P_{simuserspers}$, we group the users into clusters for extracting similar users. For clustering, we used k-means clustering technique as discussed with 25 being total number of clusters.

3.5.1 Full Cluster Approach

In full cluster approach, we derive the $P_{simuserspers}$ (similar user personalization) factor from users belonging to clusters C_j such that user u (with query q) exists in the same cluster.

$$P_{simuserspers} = \frac{\sum_{u' \in C_j \ \& \ u' \neq u} p(d|q)p(d|u')}{|C_j| - 1} \tag{3.6}$$

In equation 3.6, $|C_j|$ -1 represents the number of users in cluster C_j except the user u for whom query q is under evaluation. Now, combining equations 3.4, 3.5 and 3.6, P_{pers} can be written as 3.7.

$$P_{pers} = p(d|q) \left(\lambda p(d|u) + (1-\lambda) \frac{\sum_{u' \in C_j \ \& \ u' \neq u} p(d|u')}{|C_j| - 1} \right) \tag{3.7}$$

Equation 3.7 represents full cluster approach with simple mean average calculation. The $P_{simuserspers}$ (similar user personalization) factor can also be represented by weighted mean as in equation 3.8.

$$P_{simuserspers} = \frac{\sum_{u' \in C_j \ \& \ u' \neq u} p(d|q)p(d|u')sim(u,u')}{\sum_{u' \in C_j \ \& u' \neq u} sim(u,u')} \tag{3.8}$$

$$P_{pers} = p(d|q)\left(\lambda p(d|u) + (1-\lambda)\frac{\sum_{u' \in C_j \ \& \ u' \neq u} p(d|u')sim(u,u')}{\sum_{u' \in C_j \ \& \ u' \neq u} sim(u,u')}\right) \tag{3.9}$$

Equation 3.9 represents full cluster approach with weighted mean average calculation.

3.5.2 Partial Cluster Approach

As the numbers of users in each cluster are high, so there is a probability within a cluster to have users who are not much similar to each other. So, we decided to run the equation 3.7 with using only those users within the same cluster who are at a threshold distance from the user u under evaluation. We set the threshold distance as 0.1 (cosine similarity distance).

$$P_{pers} = p(d|q)\left(\lambda p(d|u) + (1-\lambda)\frac{\sum_{u' \in C_j \ \& \ u' \neq u \ \& \ sim(u,u') > th} p(d|u')}{N_j - 1}\right) \tag{3.10}$$

$$P_{pers} = p(d|q)\left(\lambda p(d|u) + (1-\lambda)\frac{\sum_{u' \in C_j \ \& \ u' \neq u \ \& \ sim(u,u') > th} sim(u,u')p(d|u')}{\sum_{u' \in C_j \ \& \ u' \neq u \ \& \ sim(u,u') > th} sim(u,u')}\right) \tag{3.11}$$

In equation 3.10, $N_j - 1$ are the total users from cluster C_j (which contains user u) which satisfies the two conditions; first similarity between user (u) and user (u') is greater than a specified threshold, second user (u') is not same as user (u). Equation 3.10 represents partial cluster approach with mean, while equation 3.11 represents partial cluster approach with weighted mean.

3.5.3 No Cluster Approach

In the no cluster approach, for a query q, we used all remaining users from the corpus which satisfy the threshold criteria. Basically, we didn't confine the users to a cluster.

$$P_{pers} = p(d|q)\left(\lambda p(d|u) + (1-\lambda)\frac{\sum_{\forall u' \in sim(u,u')>th \ \& \ u' \neq u} p(d|u')}{N-1}\right) \tag{3.12}$$

$$P_{pers} = p(d|q)\left(\lambda p(d|u) + (1-\lambda)\frac{\sum_{\forall u' \in sim(u,u')>th \ \& \ u' \neq u} sim(u,u')p(d|u')}{\sum_{\forall u' \in sim(u,u')>th \ \& \ u' \neq u} sim(u,u')}\right) \tag{3.13}$$

In equation 3.12, N is the total number of users which statisfies the numerator criteria; th stands for threshold value for which we consider two users as similar users. In all our runs threshold has been set to 0.1 i.e. if tf-idf similarity between two users is greater than 0.1, we consider them as similar users. Equation 3.12 represents no cluster approach with mean, while equation 3.13 represents no cluster approach with weighted mean.

In summary, for full cluster approach, we used all the users from a cluster C without any similarity threshold criteria; for partial cluster approach, we used only those users from a cluster C who satisfied a similarity threshold criterion; for no cluster approach, we used all users who satisfied a similarity threshold criterion without any clustering.

Chapter 4

DIVERSITY SEARCH

4.1 Introduction

Diversity plays an important role in modern search engines as information on World Wide Web (WWW) has increased manifold, end users don't prefer to view similar information. Also, there are lots of queries on web search which are ambiguous. For instance, a query "RSS" has different meanings for different users. For users with technical background, RSS could mean rss feeds, while for others who are interested in Indian politics, it has a different meaning as it stands for Rashtriya Swayamsevak Sangh which is an organization for a political party. Not only that, even for a technical person, a query like RSS has different aspects such as rss feeds 2.0, rss feed reader, rss tutorials etc. All these aspects can serve as subtopics for diversity retrieval in web search.

4.2 Subtopic Retrieval

Subtopic retrieval problem in information retrieval poses a different challenge compared to normal retrieval problems. It ensures that the retrieved results contain all perspectives of the query. For example, for query "algorithms" some of the potential subtopics could be sorting algorithms, search algorithms, dynamic programming algorithms, algorithms courses etc. Modern search engines attempt to resolve ambiguity by auto suggesting queries to the end users based on query logs. For example, if we enter algorithms in Google search engine, some of the auto suggestions are algorithms 4th edition, algorithms for interviews, algorithms and data structures, algorithms in java, algorithms in c, algorithms dictionary etc. It is interesting that most of the suggestions seem to be N-grams. N-gram is a collection of terms which exist adjacent to each other

in documents of a given collection. For example, for given lines "essential information that every serious programmer needs to know about algorithms and data structures. The textbook Algorithms, 4th Edition surveys the most important algorithms and data structures in use today", "algorithms and data structures" is an N-gram of 4 terms with term frequency of 2. Similarly, "Edition" is a N-gram containing 1 term with term frequency of 2.

4.2.1 Previous Work

There have been various approaches which have been suggested for subtopic retrieval such as clustering, key phrases etc. Fang et al. [6] suggest a pattern based approach. In this approach, the patterns are semantically meaning full text and are obtained by looking in the retrieved result set. Then, from the candidate patterns meaning full candidates are chosen as subtopic based on different approaches like IDF(Inverse Document Frequency), Term importance score etc. Another interesting work is presented by Zhai et al. [22] by using MMR based approach, also incorporating novelty with diversity for subtopic retrieval.

4.2.2 Proposed Approach

Our proposed approach is similar to pattern based approach as discussed in 4.2.1. Our approach consists of following steps for subtopic retrieval.

4.2.2.1 Creating N-grams

From the corpus, we make a dictionary which contains N-grams (up to n=4). N-grams are built by setting a minimum threshold N-gram frequency (n=5) and a maximum threshold frequency so that we don't have N-grams which occur frequently. Maximum Threshold frequency is set to 99 percent i.e., all N-grams which are among the top 1 percent of all N-grams based on their frequency are rejected.

4.2.2.2 Selection of candidate subtopics

A candidate selection step chooses potential subtopics candidates. In order to select candidates, we used two approaches:

Ordered start token: .In this approach, we look for N-grams whose first tokens are the same as the query first tokens and in the same order of query tokens in our dictionary. Basically, those N-grams start with query q. For example: if rss is a query then subtopic candidates are rss feeds, rss feed 2.0, rss feed reader and rss volunteers while feed reader rss is not a subtopic since the N-gram does start with the rss token.

Unordered tokens: In this approach, the order of query tokens doesn't affect the potential candidates. So, all N-grams were considered as candidates who contained all of query tokens of query q. For example, if real state housing is a query, all of the following are candidates: real state pricing housing, real state housing pricing, pricing real state housing, housing real state pricing etc.

For both the approaches discussed above, we used a Standard English stop word list to stop tokens which were present in the stop list. We first remove all stop words from each candidate and match if after removal an N-gram exists in our dictionary. If such an N-gram exists, we use the existing N-gram (stop word removed) otherwise we don't remove the stop words from the subtopic candidate.

We created a candidate table mapping potential candidates with the documents (having the potential candidates) for the query. A sample table is shown in 4.1.

In table 4.1 "kirk mccoy nc" candidate exists in documents WTX104B011368, WTX104B012134, WTX104B01497 and WTX104B01994.

4.2.2.3 Document representatives

After candidates selection, we sort the documents by their rank, where multiple representatives can exists for a given document. Among all candidates for a document,

S No.	Candidate	Documents (with candidates)
1	kirk mccoy nc	WTX104B011368, WTX104B012134, WTX104B01497, WTX104B01994
2	kirk mccoy pairing	WTX104B011192, WTX104B011193, WTX104B011232, WTX104B012012, WTX104B01439
3	kirk mccoy nc 17	WTX104B01110, WTX104B01438
4	kirk mccoy livejournal remember	WTX104B01110, WTX104B01438
5	kirk mccoy kirk spock	WTX104B011031, WTX104B011201, WTX104B011202, WTX104B012134, WTX104B013957
6	kirk mccoy kirk	WTX104B011202
7	trek xi kirk mccoy	WTX104B01110, WTX104B011202
...	...	...

Table 4.1: Candidate Selection Table

we choose the candidate as document representative with maximum tf-idf value.

$$d_r = \arg\max_{tf-idf}\{cand_1, cand_2, cand3, cand4, \ldots, cand_n\} \tag{4.1}$$

In equation 4.1, d_r represents the document representative for document d. Similarly, we have representatives for all documents which are assumed subtopic representing that document. In the next section, we discuss our greedy approach based algorithm for final subtopic selection.

4.2.2.4 Greedy based approach

Our greedy based approach helps in selection of subtopics which are diverse. We first arrange the documents (with their representative) in a set S_d by their rank so that the d_1 is the highest relevant document when the original query was used for retrieval. R represents a set which contains final subtopics to be retrieved in this approach. We first initialize R with the representative of highest document and remove the document from the set S_d. For each document in the remaining documents, we check the similarity between the document and documents whose representatives exist in set R. If there exists such a document, then we don't include the representative of

document d in final set R, meaning that aspect of subtopic has already been covered. On the other hand, if no such document exists, the current representative's perspective has not been incorporated in R so we add the representative to R. 4.2.1 defines the pseudo code for our algorithm.

4.2.2.5 Ordering unordered tokens approach

We performed this step only for Unordered tokens case as discussed above. In this step, we remove the query tokens from the final retrieved subtopic and pre-append the original query to the remaining tokens. For example, suppose the original query is brain, and retrieved subtopic is 30 Day Brain Freeze, we remove the query tokens from this subtopic so the remaining tokens are 30 Day Freeze and we append the remaining tokens to the original query so the final subtopic becomes Brain 30-Day Freeze.

4.2.3 Subtopics from Top Terms Categorized by User Judgements

The user was provided with categories related to each topic. The categories were obtained from ODP (open directories project). In this case, in order to get the subtopic, we take the group of documents from a topic in a same category, and for each document-term pair for all documents we calculate tf-idf. We choose the top tf-idf terms among all documents as our subtopics. This subtopic retrieval step was mainly done for analysis purpose.

Algorithm 4.2.1: RETRIEVESUBTOPICS(S_d)

$R \leftarrow \{R_{Sd1}\}$
$S_d \leftarrow \{S_d\} - \{S_{d1}\}$
for each $x \in S_d$
  do if $sim(d, x) > th \; \forall \; R_d \; in \; R$
  then $R \leftarrow R \; U \; \{R_{dx}\}$
return (R)

4.3 Diversity Models

The main aim of diversity models is to create re-ranking from a given initial ranking for a query, which has the maximum coverage and the minimum redundancy with respect to the different aspects underlying q. There are various diversity methods which have been proposed. For example: MMR, xQUAD etc. MMR (Maximal Marginal Relevance) model (by Jamie et al. [29])doesn't make use of subtopic for retrieval but is a greedy based approach based on document similarity. We used xQUAD diversity framework as suggested by Santos et al. [21] for diversification of results. xQUAD incorporates the probabilities of all subtopics given a topic q.

$$Score(d|q) = (1-\lambda)P(d|q) + \lambda \sum_{q_i \in Q} [P(q_i|q)P(d|q_i) \prod_{d_j \in S} (1 - P(d_j|q_i))] \tag{4.2}$$

In equation 4.2, $Score(d|q)$ is the final score after incorporating retrieved subtopics; q_i represents the i^{th} subtopic, Q represents set containing all subtopics; $P(d|q)$ provides the probability of document d given query original query q, $P(d|q_i)$ provides the probability of document d given i^{th} subtopic. S is subset which contains documents which has already been ranked. $P(d_j|q_i)$ represents probability d_j given query q_i, $P(q_i|q)$ represents the probability of subtopic q_i given query q. Results discussed in results chapter are based on the discussed xQUAD framework.

Chapter 5

PERSONALIZATION AND DIVERSIFICATION

5.1 Introduction

Personalization and diversification are two different aspects in a retrieval system. Also, for both approaches the evidence to be used is different as personalization takes feedback from user click behavior, long term query logs, and short term interests of user in a session etc., while diversification takes feedback from the ranked documents such as subtopic extraction, subtopic importance, information from above ranked documents etc. Personalization aspects are related to the individual user while diversification aims only to satisfy the maximum possible number of users. In this chapter, we propose a linear combination of both aspects of search. To our understanding, users prefer diversified results but they also want to satisfy their interests. For example, imagine a person who is an Apple shareholder with an Apple laptop using query Apple. The ideal results should cover all aspects of Apple such as current market share price, Apple store, apple fruit etc.

5.2 Previous Work

There are very few papers on personalization and diversification together. Vallet et al. [1] suggests the introduction of random variable for user u in different diversification models. Dumais et al. [13] in a patent application consider different architectures for personalization and diversification.

5.3 Approach

We used a linear combination approach for combining personalization and diversification. Because scores can be on different scales, we performed rank based normalization on the retrieved documents. For a given query q, we retrieved top n documents (d_1, d_2, ..., d_n) for both aspects. We normalize the score for document d with rank r in equation 5.1 as suggested by Vallet et al. [14].

$$N(d_r) = 1 - (\frac{r}{n}) \tag{5.1}$$

Now the normalized score for a document d with rank r is given by equation 5.2:

$$Score(d_r) = \frac{N(d_r)}{\sum_{i=1}^{n} N(d_i)} \tag{5.2}$$

We used this approach for score normalization for both personalization (discussed in chapter 3) and diversification (discussed in chapter 4). The final ranking of a given document d and query q, which considers both mentioned aspects is given by equation 5.3

$$Score_{pers-divers}(d|q) = \beta Score_{pers}(d|q) + (1 - \beta)(Score_{divers}(d|q)) \tag{5.3}$$

where β is a variable which can be tuned for both factors.

Chapter 6

EXPERIMENTS AND RESULTS

In this chapter, we discuss the results obtained from the methods discussed in previous chapters. This chapter is organized as follows. In section 6.1, we discuss about the experimental setup, in section 6.2 we discuss about the result and analysis on training queries and in section 6.3 we discuss about the results and analysis on testing queries.

6.1 Experimental Setup

6.1.1 Corpus

We used the corpus from Vallet et al. [2] comprised of user profiles and queries. The total number of users whose profile exists is around 33, where a profile represents the latest bookmarks of respective user. The bookmarked URLs by each user are then tagged using Del.icio.us, a social tagging website. This website performs automatic tagging based on URL text, but any individual can also tag an URL after reading it. So, the user profiles basically contain the tags of bookmarked pages.

The dataset also contained the URLs fetched by querying the Microsoft search engine Bing. The top 400 URLs for each query returned by the Bing API served as the documents inside the corpus. We used the Apache Nutch crawler [1] for fetching URLs to build our search corpus. Since some of the URLs were old, we were not able to fetch them because of their non existence (most of the URLs existed since this is 2012 corpus).

The query length of the queries varies from 1 to 2. Table 6.1 represents sample queries from dataset.

[1] Apache Nutch Crawler (http://apache.nutch.org)

Best run from all approaches	
Query Length	Sample Queries
1	Books, blogging, free, au, foamy, firefox, flash, camping, javascript
2	poetry education, ups power, ajax javascript, acne diet

Table 6.1: Sample Queries

6.1.1.1 Relevance Judgements

Relevance judgments have been made only for top 6 to 8 documents as ranked by Bing. We have two different judgments for each such document. One is based on personalization i.e. user judgment. These judgments are graded from 1 to 4, with 4 being highly preferred by the user and 1 being relevant but less preferred by the user. The judgments do not contain any non relevant documents but we will assume that all non judged documents are not relevant.

For diversity, the judgment file contains relevance judgments for subtopics of the query for each document. The subtopics for each query are produced by ODP (Open Directory Project). ODP is an open content directory of web links. It makes use of hierarchical ontology scheme for organizing sites. Listed websites on a similar topic are grouped into categories which can then include smaller categories. Specifically, the subtopics are the categories retrieved by ODP in response to the query. The diversity judgments are also graded judgments from 1 to 4, with 4 being highly preferred for the subtopic and 1 being less preferred for the subtopic.

6.1.2 User Representation and Clusters

Our corpus contains two different user sets. One set contains 33 users represented by tags as discussed in 3.2 while the other set contains around 73,000 users represented by query logs as discussed in 3.2.

In this section, we discuss the clusters obtained from k-means clustering (equation 3.1). Table 6.2 shows the top 10 terms in 5 different clusters. By visual inspection, all clusters seem coherent. Cluster C_1 seems to suggest users who are interested in vactions and real estate near Florida beaches; Cluster C_2 seems to suggest vehicles or

$TermRank$	Terms C_1	Terms C_2	Terms ...	Terms C_{23}	Terms C_{24}	Terms C_{25}
1	florida	parts	...	lyrics	games	business
2	new	auto	...	love	free	marketing
3	beach	ford	...	song	online	management
4	fl	car	...	myspace	game	inc
5	estate	sale	...	new	play	jobs
6	hotel	used	...	quotes	yahoo	company
7	real	truck	...	movie	kids	services
8	york	honda	...	music	disney	new
9	county	ebay	...	girl	download	group
10	hotels	cars	...	pictures	google	companies

Table 6.2: Top cluster terms

parts being purchased from commercial product search engines like eBay; Cluster C_{23} seems to suggest users interested in entertainment like movie, music, songs; Cluster C_{24} seems to suggest users interested in gaming, particular online games; Cluster C_{25} seems to suggest users interested in the business.

Figure 6.1 show the distribution of users across clusters.

6.1.3 Evaluation Measures

Evaluation measures can be categorized as follows:

6.1.3.1 Personalization Evaluation Measures

To evaluate personalization, we use standard IR evaluation measures calculated using the user judgements described in section 6.1.1.1.

Precision : It is the fraction of retrieved documents which are relevant. For example, P@k signifies the number of relevant documents in the top k retrieved documents.

$$P@k = \frac{(number\ of\ relevant\ documents\ in\ top\ k\ retrieved)}{k} \tag{6.1}$$

Any grade pf relevance is considered relevant for computing precision.

MAP (Mean Average Precision) : Average Precision is the average of the precision value obtained for the set of top k documents existing after each relevant document is retrieved, and this value is then averaged over information needs. That is, if the set of

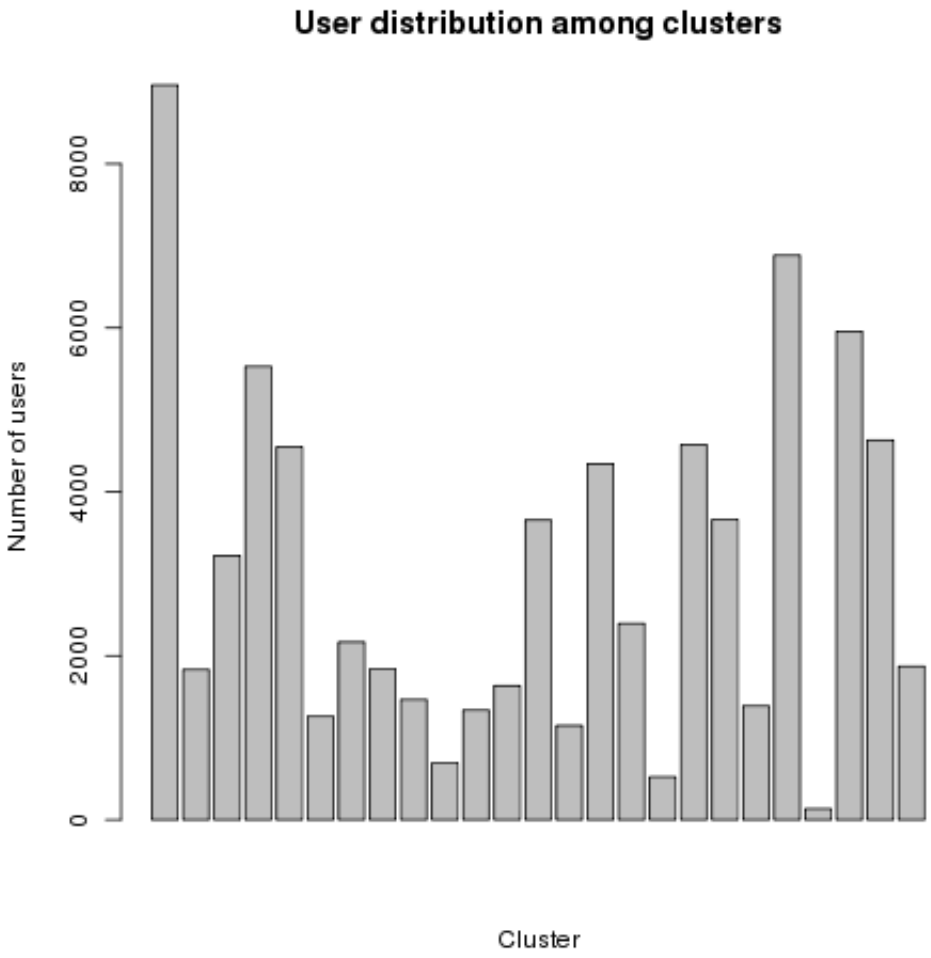


Figure 6.1: Distribution of users across all clusters

relevant documents for an information need $q \in$ Q is $d_1, \ldots, d_m$ and R_k is the set of ranked retrieval results from the top result until you get to document d_k.

$$AP(q) = \frac{1}{m}\sum_{k=1}^{m} Precision(R_k) \tag{6.2}$$

Mean average precision can be defined using equation 6.3

$$MAP(Q) = \frac{1}{|Q|}\sum_{k=1}^{|Q|} AP(q_k) \tag{6.3}$$

In equation 6.3, $|Q|$ represents the total number of queries.

DCG (Discounted Cumulative Gain) : The main aim of DCG is to penalize relevant documents which are shown lower in the ranking.

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i)} \tag{6.4}$$

Generally, a logarithmic function is used as a discount function.

nDCG (Normalized Discounted Cumulative Gain) : As DCG can vary from 0 to large values, we used nDCG. Normalized Gain is calculated by calculating IDCG (ideal DCG).

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{6.5}$$

6.1.3.2 Diversification Evaluation Measures

Following are the evaluation measures used for measurement of the diversity aspect of retrieval.

S-recall measure : S-recall stands for subtopic recall. This evaluation measure is used to measure the percentage of documents retrieved containing different subtopics. Consider a ranking $\{d_1, d_2, \ldots, d_m\}$ retrieved for topic T with n_S subtopics. This measure is defined by Zhai et al. [22].

$$S - recall\ at\ K := \frac{|\bigcup_{i=1}^{K} subtopics(d_i)|}{n_S} \tag{6.6}$$

In equation 6.6, $subtopics(d_i)$ defines the set of subtopics to which document d_i is relevant.

ERR-IA measure : ERR stands for Expected Reciprocal Rank. ERR attempts to overcome DCG measure which accounts only rank of document during evaluation, while the discount function of ERR also considers the relevance of previously ranked documents. ERR-IA is computed by calculating ERR as suggested by Chapelle et al. [10] for each subtopic, then computing a weighted average over subtopics. Equation 6.7, provides the mathematical formula for ERR-IA calculation over M subtopics where p_i is the probability of subtopic i.

$$ERR := \sum_{r=1}^{n} \frac{1}{r} P(user\ stops\ at\ position\ r)$$
$$ERR - IA = \sum_{i=1}^{M} p_i ERR_i \tag{6.7}$$

α-nDCG measure: This measure as suggested by Clarke el al. [15] assumes that each document is representative of set of subtopics. This measure rewards when documents with new subtopics are found, but also penalize document with same subtopics with the rank consideration. α is a factor which accounts for the severity of redundancy penalization, when α=0 it corresponds to standard nDCG with the number of matching subtopics for each document d used as the graded relevance value.

6.2 Results and Analysis on Training Set

We used randomly sampled 30 queries from 180 queries for training purpose of different variables. We removed 1 query (without relevance judgments) from the sampled queries.

6.2.1 Personalization Results

In this section, we will discuss about the personalization results for training queries.

Best run from all approaches					
runid	MAP	P@5	P@10	nDCG@5	nDCG@15
userscore	0.3694	0.5185	0.4852	0.3477	0.4016
full cluster (at $\lambda = 0.3$)	0.3775	0.5481	0.4963	0.3828	0.4169
partial cluster (at $\lambda = 0.9$)	0.3705	0.5111	0.4926	0.3494	0.4004
no cluster (at $\lambda = 0.9$)	0.3711	0.5111	0.4926	0.3517	0.4027

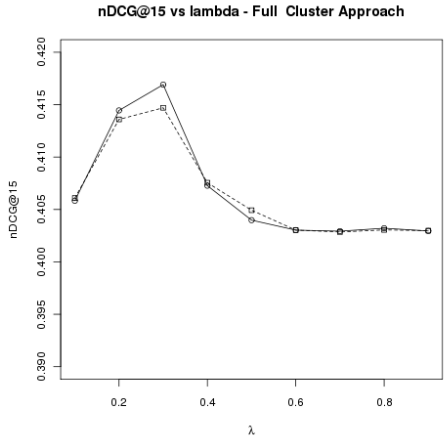
Table 6.3: Personalization Results : Best runs(based on MAP) for Training Queries

6.2.1.1 Personalization Analysis

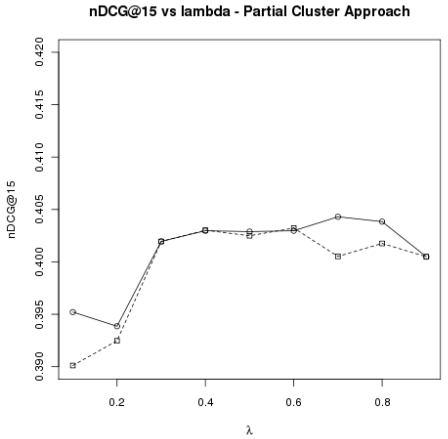
Results shown in table 6.3 suggest the use of query logs of similar users result in better performance.

As we can from figure 6.2 and figure 6.3, full cluster approach has the highest nDCG@15 and P@5. Its interesting to note as we consider users from a cluster which are at a threshold similarity to user, the nDCG values decreases. The reason could be that in case of full cluster the number of users are more and feedback term set is more compare to similar users (at threshold). Both mean and weighted mean approaches seem statistically same for most of the evaluation measures. So, similar user getting more weight doesn't seem to affect our results. P@5 for baseline i.e. using only user feedback is 0.4852 while the best P@5 happens in the same run with best nDCG@15. Also, for all approaches, nDCG@15 increases with increase in value of λ, attains maximum point, then decreases with increase in value of λ.

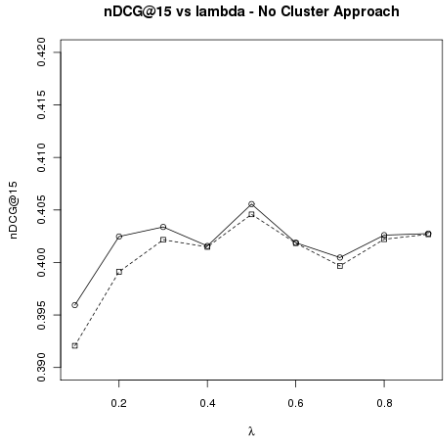
Figure 6.2 (a), (b) and (c) describes plots of nDCG@15 vs. λ for the full, partial and no cluster approach. In all three approaches, the mean approach performed nearly same to weighted mean (in dotted) as evident from the graphs (may be marginally better). Figure 6.2 (d) shows the bar plot of nDCG@5 between the best run (dotted) i.e. for full cluster approach at λ=0.3 vs. the run with only single user feedback i.e. baseline. It is interesting that our nDCG@5 is better than the baseline i.e. $P_{userpers}$ run but in terms of number of individual queries, less than 50 percent queries improved from the baseline.



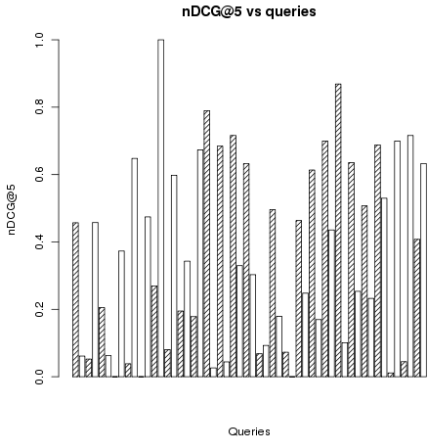
(a) nDCG@5 vs. λ for mean and weighted mean (dotted) for full cluster approach



(b) nDCG@5 vs. λ for mean and weighted mean (dotted) for partial cluster approach

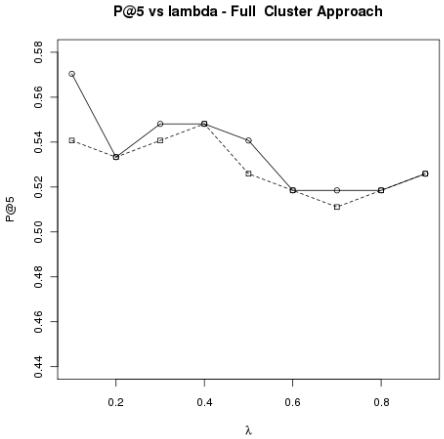


(c) nDCG@5 vs λ for mean and weighted mean (dotted) for no cluster approach

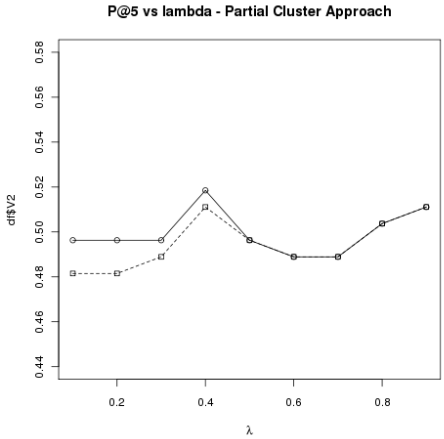


(d) nDCG5 vs. Queries for userscore (no texture) , full cluster approach with $\lambda = 0.3$ (texture)

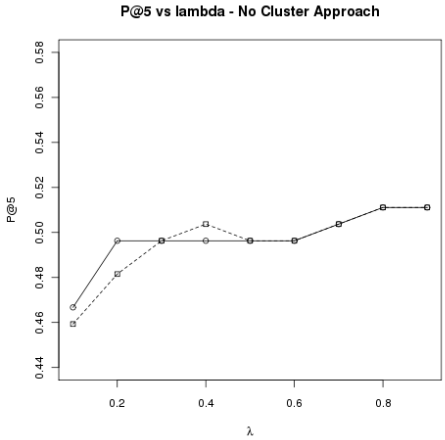
Figure 6.2: Personalization: nDCG@5 analysis plots



(a) P@5 vs. λ for mean and weighted mean (dotted) for full cluster approach



(b) P@5 vs. λ for mean and weighted mean (dotted) for partial cluster approach



(c) P@5 vs. λ for mean and weighted mean (dotted) for no cluster approach

Figure 6.3: Personalization P@5 analysis plots

6.2.2 Diversification Results

6.2.2.1 Retrieved Subtopics

Following are the results obtained using diversity methods which are proposed in 4. Below is the table which contains the retrieved subtopics for 3 queries "brain", "recipes potatoes" and "browser" by our proposed approaches.

Query	Top terms categorized	ODP categories	Unordered tokens	Ordered start tokens
brain	No Subtopics Found	brain training, brain games, brain neurosurgery , brain injuries , brain anatomy, brain definitions, brain education, brain health, brain available domains	brain cells, brain 30 day freeze, brain bug, brain exercise, brain power, brain mind, brain phi kami puzzle	brain cells, brain freeze, brain implant, brain kami, brain power, brain injury, brain game
recipes potatoes	recipes potatoes, anonymous analytics cjpberry, potato salad recipe, vowel spellcheck t.co, recipes potato sweet, solver dictionary wordnet, piwik analytics 1.8.3, potato gnocchi potatoes, conversions analytics goal, pie potatoes recipes, infoplease biographies atlas, avinash segments segmentation	recipes potatoes salad recipes, recipes potatoes pies, recipes potatoes potato recipes, recipes potatoes sweet potatoes,	recipes potatoes grand	recipes potatoes grand
browser	mmorpg 2d amaya, safari browsers browser, seo tutorial optimization, 15.0.1 firefox android, gre lookup browser9, maxthon avant opera	browser Already use, browser Useful software, browser Useful Information, browser Not relevant, browser Relavent but not useful,	browser avant, browser support, browser web browsers, browser noun, browser web, browser testing, browser based,	browser support, browser market, browser 2012, browser windows, browser testing, browser based, browser statistics
ajax javascript	ajax javascript example, ajax tutorial, ajax toolkit javascript, ajax javascript tutorial, ajax javascript library, ajax php javascript	ajax javascript tutorials, ajax javascript programming, ajax javascript social, ajax javascript definitions	ajax javascript authoring, ajax javascript 8 ways, ajax javascript rss, ajax javascript galleries slideshows, ajax javascript differences, ajax javascript statistics browser, ajax javascript accessibility	ajax javascript ticker, ajax javascript authoring, ajax javascript 8 ways, ajax javascript list, ajax javascript galleries slideshows

Table 6.4: Retrieved Subtopics for some queries

runid (at $\lambda = 0.5$)	nERR-IA@5	nERR-IA@10	nERR-IA@20
Top terms categorized	0.6199	0.5733	0.5899
ODP categories	0.4861	0.4819	0.5030
Unordered tokens	0.4110	0.4204	0.4398
Ordered start tokens	0.3782	0.3918	0.4179

Table 6.5: Diversity Results : ERR measure on training queries

runid (at $\lambda = 0.5$)	α-nDCG@5	α-nDCG@10	α-nDCG@20
Top terms categorized	0.5895	0.5245	0.5769
ODP categories	0.4727	0.4731	0.5305
Unordered tokens	0.4315	0.4484	0.5009
Ordered start tokens	0.3856	0.4126	0.4836

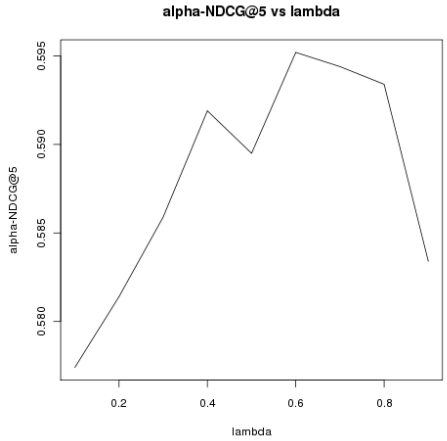
Table 6.6: Diversity Results : α-nDCG measure training queries

6.2.2.2 Diversity Analysis

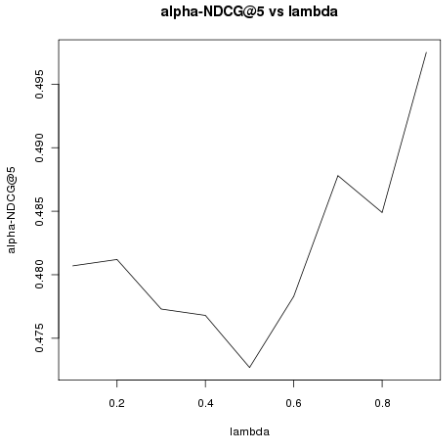
Overall our subtopic retrieval approach didn't work well. From both of the proposed approaches, our Unordered tokens based approach performed better as is evident from different measures displayed in tables 6.5, 6.6 and 6.7. It is interesting that for subtopic recall evaluation measure performed well compare to other approaches. The reason being the greedy algorithm selects the topics which are dissimilar. Figure 6.4 displays the effect of nDCG@5 vs. λ (xQUAD) over different approaches. This figure describes that increase in λ results in increase of α-nDCG values for ODP categories, Unordered tokens and Ordered start tokens. For Top terms categorized approach, the α-nDCG values increases with increase in λ upto a point, then decreases.

6.3 Results and Analysis on Test Set

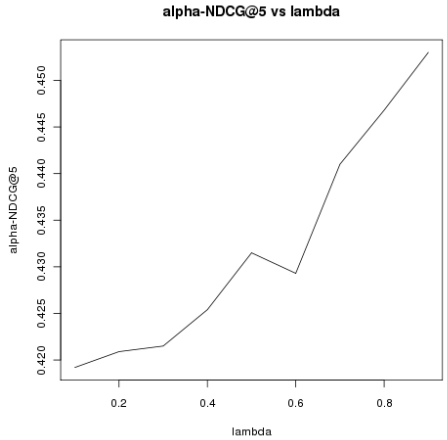
We used the training queries for training the optimum values of different variables before we apply both personalization and diversification. We obtained the test queries by random sampling of 25 queries. As all queries dont have relevance judgments, so we dropped 2 queries from 25 queries.



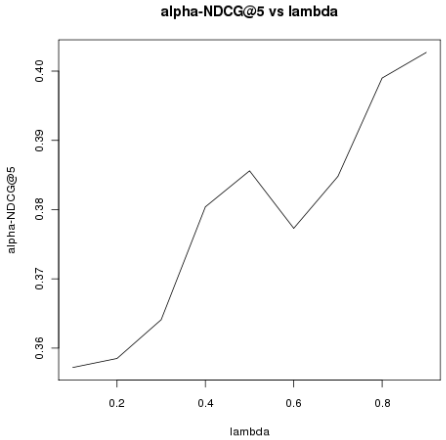
(a) α-nDCG@5 vs. λ (xquad diversity) for top terms categorized run



(b) α-nDCG@5 vs. λ (xquad diversity) for ODP categories run



(c) α-nDCG@5 vs. λ (xquad diversity) for Unordered tokens run



(d) α-nDCG@5 vs. λ (xquad diversity) for Ordered start tokens run

Figure 6.4: α-nDCG@5 vs. λ for different runs

runid (at $\lambda = 0.5$)	strec@5	strec@10	strec@20
Top terms categorized	0.4129	0.4865	0.6050
ODP categories	0.3500	0.4857	0.6196
Unordered tokens	0.3740	0.5061	0.6246
Ordered start tokens	0.3077	0.4538	0.6298

Table 6.7: Diversity Results : strec measure training queries

6.3.1 Personalization and Diversification

We study this problem of diversification on different diversity approaches as suggested above. As above, the best personalization run was based on full cluster approach, we use the same.

Results shown in table 6.8 incorporates the user factor of equation 3.4 on test queries with.

User Score					
runid	map	P@5	P@10	nDCG@5	nDCG@15
userscore	0.3356	0.5182	0.4864	0.3576	0.4124
full cluster (at $\lambda = 0.3$)	0.3367	0.5182	0.4818	0.3662	0.4242
partial cluster (at $\lambda = 0.9$)	0.3705	0.5111	0.4926	0.3494	0.4004
no cluster (at $\lambda = 0.9$)	0.3711	0.5111	0.4926	0.3517	0.4027

Table 6.8: Personalization Results for Test Queries

6.3.1.1 Top terms categorized and Full cluster approach

We combine both "Top terms categorized" and "Full cluster approach" using the equation 5.3, with different values of β. Table 6.9 shows the effect of variation between personalization and diversification. Figure 6.5 displays the behavior of nDCG@5 and α-ndcg@5.

Further, in table 6.9 the product of nDCG@10 and α-ndcg@10 varies with the factor β suggests that a stage can be reached are optimum for users (peak in graph) without much penalized from both diversity and personalization.

Another important factor to observe is that with $\beta = 1.0$ i.e. with just personalization nDCG@5 is 0.3819 which is greater than nDCG@5 from table 6.8 i.e. 0.3576.

β	nDCG@5	nDCG@10	α-nDCG@5	α-nDCG@10	HM(nDCG@10, α-nDCG@10)
1.0	0.3819	0.3947	0.5157	0.5206	0.4490
0.9	0.3740	0.3884	0.5344	0.5349	0.4500
0.8	0.3495	0.3829	0.5414	0.5391	0.4478
0.7	0.3687	0.3866	0.5950	0.5506	0.4543
0.6	0.3408	0.3777	0.5965	0.5522	0.4486
0.5	0.3228	0.3635	0.5972	0.5587	0.4404
0.4	0.3079	0.3473	0.6092	0.5633	0.4297
0.3	0.3072	0.3332	0.6205	0.5762	0.4222
0.2	0.2999	0.3289	0.6343	0.5766	0.4189
0.1	0.2673	0.3099	0.6061	0.5771	0.4033
0.0	0.2629	0.2948	0.6105	0.5756	0.3899

Table 6.9: Personalization Diversification : nDCG and α-nDCG Measure for combined Top terms categorized and Full cluster approach

6.3.1.2 ODP categories and Full cluster approach

We combine both ODP categories and Full cluster approach using the equation 5.3, with different values of β. Table 6.10 shows the effect of variation between personalization and diversification. Figure 6.6 displays the behavior graphically. Its interesting to see that as β decreases i.e. personalization decreases, the diversity also decrease. The main reason behind this failure is our diversity subtopics retrieval models.

β	nDCG@5	nDCG@10	α-nDCG@5	α-nDCG@10	HM(nDCG@10, α-nDCG@10)
1.0	0.3819	0.3992	0.5157	0.5215	0.4522
0.9	0.3784	0.3988	0.5286	0.5367	0.4576
0.8	0.3748	0.3942	0.5482	0.5269	0.4510
0.7	0.3828	0.3849	0.5577	0.5268	0.4448
0.6	0.3598	0.3626	0.5516	0.5263	0.4294
0.5	0.3395	0.3644	0.5487	0.5279	0.4312
0.4	0.3294	0.3567	0.5470	0.5239	0.4244
0.3	0.3187	0.3398	0.5219	0.5093	0.4076
0.2	0.2987	0.3318	0.4963	0.4999	0.3989
0.1	0.2876	0.3222	0.4956	0.4929	0.3897
0.0	0.2811	0.3142	0.4847	0.4826	0.3806

Table 6.10: Personalization Diversification : nDCG and α-nDCG Measure for combined ODP categories and Full cluster approach

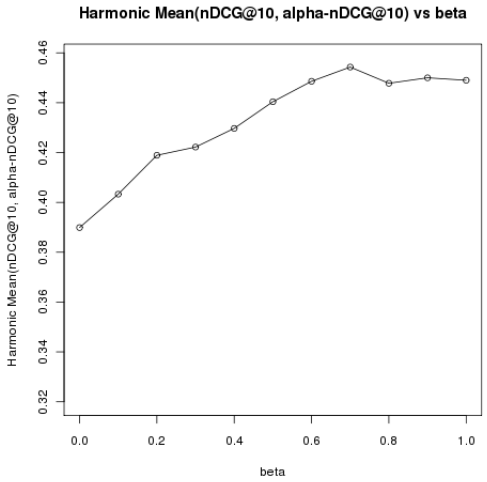


Figure 6.5: Personalization Diversification : Harmonic Mean (nDCG@10, α-nDCG@10) vs β for combined Top terms categorized and Full cluster approach

Results Comparison : In this paragraph, we compare our results with the xQUAD approach proposed by David et al. [2]. Table 6.11 suggests that methods

Run	P@5	nDCG@5	α-nDCG@5
xQUAD	0.6700	0.3200	0.7930
Full Cluster Approach + ODP($\beta = 1$)	0.5182	0.3819	0.5157

Table 6.11: Result Comparison

proposed by David et al. [2] does well in most of the evaluation measures.The reason of our below performance could be slightly different dataset as we crawled the dataset later. So, some of the URLs might have got changed.

6.3.1.3 Unordered tokens and Full cluster approach

We combine both Unordered tokens and Full cluster approach using the equation 5.3, with different values of β. Table 6.12 shows the effect of variation between personalization and diversification. Figure 6.7 displays the behavior which is same for combination of ODP categories and Full cluster approach.

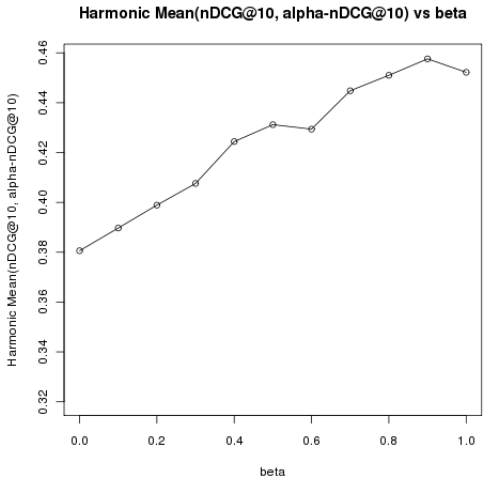


Figure 6.6: Personalization Diversification : Harmonic Mean (nDCG@10, α-nDCG@10) vs β for combined ODP categories and Full cluster approach

6.3.1.4 Ordered start tokens and Full cluster approach

"Ordered start tokens" and "Full cluster approach" case behaves similar to the Ordered start tokens and Full cluster approach as shown in table 6.13 and figure 6.8.

β	nDCG@5	nDCG@10	α-nDCG@5	α-nDCG@10	HM (nDCG@10, α-nDCG@10)
1.0	0.3819	0.3921	0.5157	0.5186	0.4466
0.9	0.3887	0.3867	0.5352	0.5264	0.4459
0.8	0.3691	0.3891	0.5418	0.5220	0.4459
0.7	0.3748	0.3882	0.5305	0.5203	0.4446
0.6	0.3699	0.3700	0.5172	0.5164	0.4311
0.5	0.3426	0.3590	0.5287	0.5185	0.4243
0.4	0.3381	0.3513	0.5263	0.5179	0.4186
0.3	0.3140	0.3354	0.5086	0.5051	0.4031
0.2	0.3269	0.3354	0.5208	0.5084	0.4042
0.1	0.2541	0.3062	0.4708	0.4668	0.3698
0.0	0.2493	0.3798	0.4608	0.2978	0.3338

Table 6.12: Personalization Diversification : nDCG and α-nDCG Measure for combined Unordered tokens and Full cluster approach

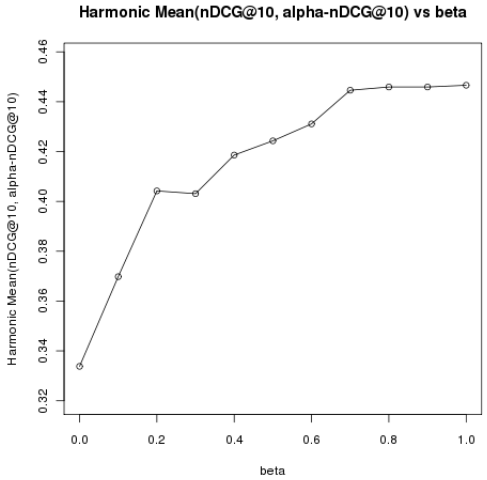


Figure 6.7: Personalization Diversification : Harmonic Mean (nDCG@10, α-nDCG@10) vs β for combined Unordered tokens and Full cluster approach

β	nDCG@5	nDCG@10	α-nDCG@5	α-nDCG@10	HM (nDCG@10, α-nDCG@10)
1.0	0.3819	0.3926	0.5157	0.5162	0.4460
0.9	0.3819	0.3926	0.5267	0.5316	0.4516
0.8	0.3677	0.3824	0.5447	0.5364	0.4465
0.7	0.3559	0.3631	0.5217	0.5239	0.4289
0.6	0.3298	0.3548	0.5336	0.5242	0.4232
0.5	0.3078	0.3345	0.5389	0.5264	0.4091
0.4	0.2737	0.3028	0.5053	0.4874	0.3735
0.3	0.2549	0.2956	0.4933	0.4808	0.3661
0.2	0.2470	0.2746	0.4748	0.4602	0.3440
0.1	0.2198	0.2621	0.4456	0.4485	0.3309
0.0	0.2112	0.2571	0.4418	0.4483	0.3268

Table 6.13: Personalization Diversification : nDCG and α-nDCG Measure for combined Unordered tokens and Full cluster approach

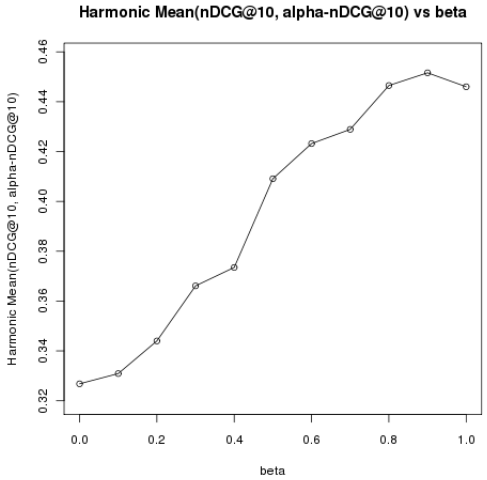


Figure 6.8: Personalization Diversification : Harmonic Mean (nDCG@10, α-nDCG@10) vs β for combined Ordered start tokens and Full cluster approach

Chapter 7

CONCLUSION

In this chapter, we conclude all our runs for both aspects.

7.1 Personalization

We attempted 3 approaches for personalization, full cluster approach, partial cluster approach and no cluster approach. In full clustering approach, all users within a cluster are considered as similar for feedback about the interests of users u under evaluation. In partial clustering approach, users within a cluster (which belongs to user u) at a threshold distance are considered similar users. In no clustering approach, we consider all remaining users within a threshold distance (0.1) as similar users. The main idea is to use the interests of similar users to predict information about user u. Full clustering approach seems to work well with some restrictions (which require study) since the bar graph for nDCG@10 suggests considerable improvement but only for less than 50 percent queries.

7.2 Diversification

We attempted 2 approaches, Unordered tokens and Ordered start token. Both approaches don't seem to work well. Out of both Unordered tokens is a better approach as it doesn't restrict N-grams to be in the order of query terms. Only evaluation measure where Unordered tokens approach did better (even better than ODP) is subtopic recall which is interesting. The other approach i.e. top terms categorized, is mainly attempted for a test purpose since all the other 3 approaches are suggesting lower alpha-nDCG value than baseline.

7.3 Personalization and Diversification

Since our full based clustering approach worked better, we used this approach with all diversity approaches. For top terms categorized approach, results suggest that a β value can found around 0.7 where both personalization and diversification (without much) both can be accommodated in a search model since access of any of the aspects will result in great dissatisfaction to users. For all other 3 combination the β value is around 1.0 which suggest that the personalization can also incorporate diversity or personalization doesn't obstruct diversity in great extent.

BIBLIOGRAPHY

[1] David Vallet. Personalized Diversification of Search Results. *In SIGIR '12 Proceedings*, 2012.

[2] David Vallet. Personalizatin and Diversification Dataset. *http://ir.ii.uam.es/dvallet/persdivers/index.htm*, 2012.

[3] Jaime Teevan, Susan Dumais and Eric Horvitz. Potential for Personalization. *ACM Transactions on Computer-Human Interaction*, Volume 17 Issue 1, March 2010 Article No. 4.

[4] Hang Cui, Ji-Rong Wen, Jian-Yun Nie and Wei-Ying Ma. Query Expansion by Mining User Logs. *IEEE Transactions on knowledge and data engineering*, Vol. 15, No. 4, July/August 2003.

[5] Donald Metzler and W. Bruce Croft. A Markov Random Field Model for Term Dependencies. *SIGIR '05 Proceedings of the 28th annual international ACM SIGIR conference*, Pages 472-479.

[6] Wei Zheng, Xuanhui Wang, Hui Fang and Hong Cheng, An Exploration of Pattern-based Subtopic Modeling for Search Result Diversication*JCDL '11 Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries* Pages 387-388.

[7] Joon Ho Lee. Analyses of multiple evidence combination. *In SIGIR '97*, pages 267-276, 1997.

[8] R. White, P.N. Bennett and S. Dumais. Predicting Short-term Interests Using Activity-Based Search Contexts. *In Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10)* Toronto, Canada. Oct 2010.

[9] David Sontag, Kevyn Collins-Thompson, Paul N. Bennett, Ryen White, Susan Dumais and Bodo von Billerbeck. Probabilistic Models for Personalizing Web Search. *In Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM '12)* Seattle, Washington, February 2012.

[10] Olivier Chapelle, Donald Metlzer, Ya Zhang and Pierre Grinspan. Expected reciprocal rank for graded relevance. *CIKM '09 Proceedings of the 18th ACM conference on Information and knowledge management*, Pages 621-630.

[11] Ji-Rong Wen, Jian-Yun Nie and Hong-Jiang Zhang. Clustering User Queries of a Search Engine. *ACM Transactions on Information Systems (TOIS)*, Volume 20 Issue 1, Pages 59 - 81, January 2002.

[12] Susan Dumais, G. Buscher and E. Cutrell. Individual differences in gaze patterns for Web search. *In Proceedings of IIiX*, 2010.

[13] F. Radlinksi and Susan Dumais. Improving personalized web search using results diversification. *In Proceedings of SIGIR'06*, 2006, 691-692.

[14] Miriam Fernndez, David Vallet and Pablo Castells. Probabilistic Score Normalization for Rank Aggregation. *In proceedings of ECIR'06*, 2006.

[15] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Bttcher and Ian MacKinnon, Novelty and Diversity in Information Retrieval Evaluation. *SIGIR '08 Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, Pages 659-666, 2008.

[16] Filip Radlinski, Susan Dumais and Eric J. Horvitz. Diversifying search results for improved search and personalization. *Patent - US7761464*, Issue date Jul 20, 2010.

[17] Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. *SIGIR '06 Proceedings of the 29th annual international ACM SIGIR*, Pages 691 - 692 , 2006.

[18] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher and Robin Burke. Personalized Recommendation in Social Tagging Systems Using Hierarchical Clustering. *RecSys '08 Proceedings of the 2008 ACM conference*, Pages 259-266, 2008.

[19] Andreas Hotho, Robert Jaschke, Christoph Schmitz and Gerd Stumme. Information Retrieval in Folksonomies: Search and Ranking. *ESWC'06 Proceedings of the 3rd European conference on The Semantic Web*, Pages 411-426, 2006.

[20] Praveen Chandar and Ben Carterette. Analysis of Various Evaluation Measures for Diversity. *In proceedings of ECIR'11*, 2011.

[21] Rodrygo L. T. Santos, Craig Macdonald and Iadh Ounis. Exploiting Query Reformulations for Web Search Result Diversication. *WWW '10 Proceedings of the 19th international conference on World wide web*, Pages 881-890 , 2010.

[22] Cheng Xiang Zhai, William W. Cohen and John Lafferty. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. *SIGIR '03 Proceedings of the 26th annual international ACM SIGIR conference*, Pages 10-17, 2003.

[23] Tao Xiang and Shaogang Gong. Spectral clustering with eigenvector selection. *Pattern Recognition Journal*, Pages 1012-1029, Volume 41 Issue 3, March, 2008.

[24] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman and Angela Y. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, Volume: 24 , Issue: 7 Page(s): 881 - 892, 2002.

[25] Jiahui Liu, Peter Dolan and Elin Rnby Pedersen. Personalized news recommendation based on click behavior. *IUI '10 Proceedings of the 15th international conference on Intelligent user interfaces*, Pages 31-40, 2010.

[26] Hang Li. A Short Introduction to Learning to Rank. *IEICE Transactions on Information and Systems* Vol.E94-D No.10 pp.1854-1862.

[27] Jimmy Lin. A Hadoop toolkit for web-scale information retrieval research. *http://lintool.github.com/Ivory/*

[28] Apache Nutch Crawler. *http://nutch.apache.org/*.

[29] Jamie Carbonell and Jade Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. *Proceedings of the 21st annual international ACM SIGIR conference*, 1998.