M. Tech. (Computer Science) Dissertation Series

Understanding the Performance of Expanded Queries

A dissertation submitted in partial fulfilment of the requirements for the M. Tech. (Computer Science) degree of the Indian Statistical Institute.

By

Soumajit Pramanik

under the supervision of

Dr. Mandar Mitra

INDIAN STATISTICAL INSTITUTE

203, Barackpore Trunk Road
Calcutta - 700 108

Indian Statistical Institute

203, B.T. Road. Kolkata : 700108

CERTIFICATE

I certify that I have read the thesis titled **"Understanding the Performance of Expanded Queries"**, prepared under my guidance by Soumajit Pramanik, and in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master of Technology in Computer Science of the Indian Statistical Institute.

Mandar Mitra
Associate Professor
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute

Kolkata
July, 2013.

Declaration of Authorship

I, Soumajit Pramanik, declare that this thesis titled, "Understanding the Performance of Expanded Queries" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____

Date: ____

Acknowledgements

To Prof. Mandar Mitra whose expert guidance, infinite patience and constant encouragement has endowed me with a strong foundation in Information Retrieval and made this dissertation possible. Words are not enough to describe his contribution to my work.

To Mrs Dipasree Pal who was always available to answer all my queries and helped me in using the Terrier tool. She also supplied me a huge collection of differnt variants of expanded queries which was one of the most important prerequisite resources for my work.

Soumajit Pramanik
July, 2013.

Contents

List of Figures

List of Tables

Abbreviations

DFR	**D**ivergence **F**rom **R**andomness
DFO	**D**ynamic **F**eedback **O**ptimization
IEQ	**I**deal **E**xpanded **Q**uery
IR	**I**nformation **R**etrieval
KLD	**K**ullback **L**eibler **D**ivergence
LCA	**L**ocal **C**ontext **A**nalysis
MAP	**M**ean **A**verage **P**recision
QE	**Q**uery **E**xpansion
RF	**R**elevance **F**eedback
RBLM	**R**elevance **B**ased **L**anguage **M**odel
TF	**T**erm **F**requency
IDF	**I**nverse **D**ocument **F**requency

Chapter 1

Introduction

Information Retrieval hardly needs any introduction today. Surveys show that about 85% of the users of the internet use popular interactive search engines to satisfy their information need. Such is the impact of information retrieval, particularly search engines, in our daily lives that the word *google* has been added to the Oxford English Dictionary as a verb, whereby *Google it* now means *search it*!

1.1 Brief Introduction To Information Retrieval

Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources [2]. Searches can be based on metadata or on full-text (or other content-based) indexing.

Definition 1.1. Information retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers) [12].

1.1.1 Document, Collection And Query

A *document* is a file containing significant text content. It has some minimal structures e.g. title, author, date, subject etc.. Examples of documents are web pages, email, books, news, stories, scholarly papers, text messages, MSWord documents, MSPowerpoint documents, PDF documents, forum postings, blogs etc.

A set of similar documents is called *collection.* Generally all activities of an IR system is performed on a collection of documents with a pre-defined structure or format (e.g. normal text file, pdf, MSWord etc.).

An information retrieval process begins when a user enters a *query* into the system. Queries are formal statements of information needs, for example search strings in web search engines. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy.

1.1.1.1 Vector Space Model

In vector space model documents and queries are represented as vectors. This model is very commonly used.

$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$

$$q = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$$

Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. Several different ways of computing these values, also known as (term) weights, have been developed. One of the best known schemes is tf-idf weighting [3]. TF means term frequency which is the no. of times a term occurs in a document. IDF means inverse document frequency which is the log of the ratio of collection size and no. of documents containing that term.

The definition of term depends on the application. Typically terms are single words, keywords, or longer phrases. If the words are chosen to be the terms, the dimensionality of the vector is the number of words in the vocabulary (the number of distinct words occurring in the corpus).

Vector operations can be used to compare documents with queries.

1.1.2 Retrieval And Evaluation Procedure

1.1.2.1 Retrieval

Before actual retrieval begins,the documents within the collection must be indexed. Indexing involves processing each document in a collection and building a data structure of indexed documents. Following are the steps of indexing:

1. Reading and parsing a document.
2. Stopword removal and stemming of each term in the document [14].
3. Inserting each term in the data structure of indexed documents.

The first step of retrieval is to assign a score to each document according to its relevance with the given query. For this generally inner product similarity is used.

The inner product similarity between document dj and query q can be calculated as:

$$\mathrm{sim}(d_j, q) = \frac{\mathbf{d_j} \cdot \mathbf{q}}{\|\mathbf{d_j}\| \, \|\mathbf{q}\|} = \frac{\sum_{i=1}^{N} w_{i,d_j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,d_j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}}$$

According to this similarity the documents are sorted and the top K documents are returned to the user. Value of K varies from system to system. Figure 1.1 gives an overview of the procedure.

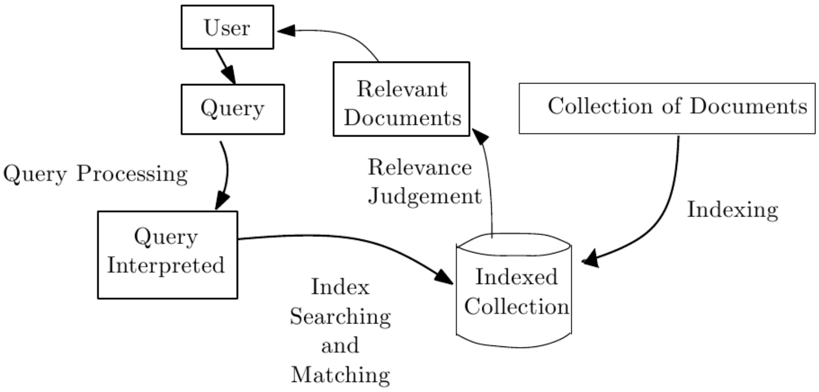

Figure 1.1: Outline of IR Procedure.

1.1.2.2 Evaluation

1. Evaluation of Unranked Retrieval Sets:

   The two most basic parameters for performance measurement of an IR system are *precision* and *recall.* These are initially defined for the simple case where the IR system returns only a set of documents. These definitions can be extended for IR systems which returns a set of documents along with ranks.

   Precision is the fraction of the documents retrieved that are relevant to the user's information need.

   $$Precision = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}}$$

   Recall is the fraction of the documents that are relevant to the query that are successfully retrieved.

   $$Recall = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents in the collection}}$$

2. Evaluation of Ranked Retrieval Results:

   The ranked retrieval results are now standard with search engines. In a ranked retrieval context, appropriate sets of retrieved documents are naturally given by the top K retrieved documents.

   In recent years, *Mean Average Precision(MAP)* has become a standard parameter [11]. It has been shown that MAP has especially good discrimination and stability among evaluation measures.

   The concept of *Average Precision*, henceforth abbreviated as AP, is required to define MAP. For a single query, AP is the average of the precision values obtained for the set of top K documents existing after each relevant document is retrieved. For MAP, such AP values are then averaged over all information needs.

   Let the set of documents retrieved for a query q_j be $D = \{d_1, \dots d_{m_j}\}$ such that document d_i has rank i. Let R_j be the set of all documents that are relevant to q_j and rel_{jk} be an indicator variable which is 1 if $d_k \in R_j$, 0 otherwise. Let $P(i)$ be the precision of the first i documents in D. Then, Average Precision for query q_j is defined as:

$$AP_{q_j} = \frac{1}{|R|}\sum_{i=1}^{m_j} P(i) \cdot rel_{ji}$$

Let the query set be Q. MAP of Q is the average of AP_{q_j} for all $q_j \in Q$. So,

$$MAP = \frac{1}{|Q|}\sum_{j=1}^{|Q|} AP_{q_j}$$

1.1.3 Vocabulary Problem

The most critical language issue for retrieval effectiveness is the term mismatch problem: the indexers and the users do often not use the same words. This is known as the vocabulary problem [5] [6], compounded by synonymy (same word with different meanings, such as 'java') and polysemy (different words with the same or similar meanings, such as 'tv' and 'television'). Synonymy, together with word inflections (such as with plural forms, "television" versus "televisions"), may result in a failure to retrieve relevant documents, with a decrease in recall. Polysemy may cause retrieval of erroneous or irrelevant documents, thus implying a decrease in precision. This problem is particularly acute in the case of short queries.

To deal with the vocabulary problem, several approaches have been proposed including interactive query refinement, relevance feedback, word sense disambiguation, and search results clustering. One of the most natural and successful techniques is to expand the original query with other words that best capture the actual user intent, or that simply produce a more useful query, i.e. a query that is more likely to retrieve relevant documents. This approach is known as Query Expansion.

1.2 Query Expansion

The aim of query expansion (QE) is to reduce this query-document mismatch by adding related or synonymous words or phrases to the query (these words/phrases are called expansion terms).

For example, a user may be looking for information on "atmospheric pollution from automobile emissions", and a relevant document may discuss pollution arising out of smoke emitted by motor vehicles. Thus, it may be useful to add the terms cars and motor vehicles to the example query given above. This is precisely the function of QE algorithms.

1.3 Our Work

1.3.1 Motivation Behind Work

There exists several standard Automatic Query Expansion algorithms like KLD [8], RBLM [10], LCA [16], DFR [4] etc. Each of them has its own term selection and term weighting strategy. The relative performances of these algorithms vary over different input queries. Also there are queries for which performance does not improve, or even worsens, when it is expanded.

The actual reasons behind varying relative performances of AQE algorithms have not really been searched carefully. But understanding the reason as to why some query expansion algorithms perform better than other query expansion algorithms for a given query is quite important. For a given input query if the expansion algorithm A_i performed better than algorithm A_j, then what did algorithm A_i do that algorithm A_j did not, and which resulted in A_i performing better? Similarly if all expansion algorithms perform poorly for a given query, where is it that all of them are going wrong?

These are the questions whose answer may shed considerable light on how to formulate better expanded queries, and these are precisely the questions that we try to answer in this thesis.

1.3.2 Problem Definition

1.3.2.1 Hypothesis

For any query, there exists a set of good expansion terms and if we include them in the expanded query with optimal weights, a very high (0.85-1) MAP can be achieved.

Now suppose for a query Q that set of good terms is S. Let there be two query expansion algorithms A_1 and A_2 which take the query Q as input and expand it into Q_1 and Q_2 respectively.

Our hypothesis is that,

1. if Q_1 contains more terms from S than Q_2 and has assigned them good weights, then performance of Q_1 must be better than Q_2;
2. conversely, if Q_1 is found to perform better than Q_2, then it must have effectively picked up more terms from S and assigned them better weights than Q_2.

To test this hypothesis, at first we have to select a set of queries and a set of standard query expansion algorithms. For each such query expansion method, we may consider different variations of them (variations based on parameters like no. of terms etc). For our case we have chosen the TREC 8 collection (queries 401-450) [1]. The chosen query expansion algorithms are DFR, KLD, LCA, LCANEW [13] and RBLM. We considered 45 variations of each of them (total 225).

1.3.2.2 Problem Statement

Our problem is basically threefold.

1. First, we have to find a ranked set of good terms for each query. This step is called "Ideal expanded query (IEQ) generation". For this, all the available relevance information can be exploited. For every such IEQ, we have to achieve a MAP value as close to 1 as possible.

   This work has been entirely done by Snehasish Mukherjee and it is clearly described in his M.Tech Thesis [12]. The IEQs generated by his algorithm are able to achieve an average MAP of 0.9 which is quite sufficient for our current work. So, we have not applied any modification on his algorithm.

2. For a query $Q^{(i)}$ from the test collection (i $\in$ 401,. . .,450), let $Q_1^{(i)}$,. . .,$Q_{225}^{(i)}$ be the different versions of this query. Our next step is to calculate the overlap/similarity between $IEQ^{(i)}$ and $Q_1^{(i)}$,. . .,$Q_{225}^{(i)}$ $\forall$ i $\in$ 401,. . .,450. We use $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)})$

to denote the similarity or overlap between the ideal expanded form of $Q^{(i)}$ and $Q_j^{(i)}$. A number of different functions are used as Sim_{Query}. These are described in detail in Chapter 2. We then rank $Q_j^{(i)}$ 1 $\leq$ j $\leq$ 225 according to $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)})$.

3. Finally, we compare the actual ranking of queries (based on their performance, i.e. MAP values) with the predicted ranking of queries (based on Sim_{Query}). We use Sim_{Ranks} to denote the rank correlation function used to compare these two rankings. Three standard rank correlation measures Pearson correlation coefficient, Kendall rank correlation coefficient and Spearman rank correlation coefficient are used for Sim_{Ranks}. These are briefly described in Chapter 2.

In this way, we can systematically confirm our hypothesis by looking at the similarity of terms selected by different QE techniques with the terms of the IEQ as a predictor of the final performance of those QE techniques.

1.4 Organization of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2 we introduce different existing and new correlation metrics. We also report the results obtained for all of them and try to analyse those results in the same Chapter. Finally we conclude the thesis in Chapter 3 by summarizing our findings and providing pointers to the future direction of this research.

Chapter 2

Correlation Metrics

Recall from Section 1.3.2.2 that we use $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)})$ to denote the similarity or overlap between the ideal expanded form of $Q^{(i)}$ and $Q_j^{(i)}$. In this chapter, we briefly describe various functions that we use as Sim_{Query}.

2.1 Existing Metrics

2.1.1 Jaccard Index

This is a very basic metric which measures the similarity between two lists by calculating the ratio of no. of intersecting terms and total no. of terms (i.e. union size).

Jaccard Index: $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \frac{\mid IEQ^{(i)} \cap Q_j^{(i)} \mid}{\mid IEQ^{(i)} \cup Q_j^{(i)} \mid}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are represented in set notation.

2.1.2 Inner Product Similarity

This is the dot product of the term-weights from both the lists. We can normalize the dot product in various ways:

2.1.2.1 L_1 Norm

Here normalization is done by dividing the dot product by the product of the sum of all term-weights from both the lists.

Inner Product Similarity (L_1 Norm): $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \frac{IE\vec{Q}^{(i)}.\vec{Q}_j^{(i)}}{\sum_n w_n^{IEQ^{(i)}} \cdot \sum_p w_p^{Q_j^{(i)}}}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are represented in vector notation. $w_n^{IEQ^{(i)}}$ is the weight of the n^{th} term of $IEQ^{(i)}$ and $w_p^{Q_j^{(i)}}$ is the weight of the p^{th} term of $Q_j^{(i)}$.

2.1.2.2 L_2 Norm

Here normalization is done by dividing the dot product by the product of Euclidian norms of both the lists.

Inner Product Similarity (L_2 Norm): $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \frac{IE\vec{Q}^{(i)}.\vec{Q}_j^{(i)}}{\|IE\vec{Q}^{(i)}\| \cdot \|\vec{Q}_j^{(i)}\|}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are represented in vector notation.

2.1.2.3 L_∞ Norm

Here normalization is done by dividing the dot product by the product of maximum term-weights from both the lists.

Inner Product Similarity (L_∞ Norm): $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \frac{IE\vec{Q}^{(i)}.\vec{Q}_j^{(i)}}{max\Big(IE\vec{Q}^{(i)}\Big) \cdot max\Big(\vec{Q}_j^{(i)}\Big)}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are represented in vector notation.

2.1.2.4 Unnormalized

Here no normalization is done.

Inner Product Similarity (Unnormalized): $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = IE\vec{Q}^{(i)}.\vec{Q}_j^{(i)}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are represented in vector notation.

2.1.3 NDCG

NDCG is an abbreviation for Normalized Discounted Cumulative Gain [7] [15]. This metric is generally used for evaluating the performance of an Information Retrieval system [11].

We used it in measuring similarity between an expanded query $Q_j^{(i)}$ and the corresponding Ideal expanded query $IEQ^{(i)}$ assuming that the terms belonging to the $IEQ^{(i)}$ are relevant terms and their weights are the corresponding gains received by the expanded query if it includes them.

The Discounting function reduces the achieved gain on the basis of rank. The discount increases with the intersecting term's rank in $Q_j^{(i)}$. Generally logarithmic discounting function is used. The following is one standard formulation of NDCG.

NDCG: $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \sum_{t \in IEQ^{(i)} \, \cap \, Q_j^{(i)}} \frac{2^{W_t^{IEQ^{(i)}}} - 1}{log(1 + R_t^{Q_j^{(i)}})}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are represented in set notation, $W_t^{IEQ^{(i)}}$ is the weight of term t in $IEQ^{(i)}$ and $R_t^{Q_j^{(i)}}$ is the rank of the term t in $Q_j^{(i)}$.

2.1.4 Pearson Correlation Coefficient

Pearson Correlation Coefficient(r):

$$Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \frac{\sum_n \left(w_n^{IEQ^{(i)}} - w^{I\bar{E}Q^{(i)}}\right)\left(w_n^{Q_j^{(i)}} - w^{\bar{Q}_j^{(i)}}\right)}{\sqrt{\sum_n \left(w_n^{IEQ^{(i)}} - w^{I\bar{E}Q^{(i)}}\right)^2 \left(w_n^{Q_j^{(i)}} - w^{\bar{Q}_j^{(i)}}\right)^2}}$$

where $w_n^{IEQ^{(i)}}$ and $w_n^{Q_j^{(i)}}$ are the weights of the n^{th} terms in the ideal and candidate queries. $w^{I\bar{E}Q^{(i)}}$ and $w^{\bar{Q}_j^{(i)}}$ are the means of term-weights of $IEQ^{(i)}$ and $Q_j^{(i)}$. Here only terms belonging to both $IEQ^{(i)}$ and $Q_j^{(i)}$ are considered.

2.1.5 Kendall Rank Correlation Coefficient (τ)

Kendall rank correlation coefficient (τ):

$$Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \frac{\text{no of concordant pairs} - \text{no of discordant pairs}}{\frac{1}{2}n(n-1)}$$

where a *concordant pair* is a pair of terms in $IEQ^{(i)} \cap Q_j^{(i)}$ whose relative order in $Q_j^{(i)}$ is the same as that in $IEQ^{(i)}$. Similarly a *discordant pair* is a pair of terms in $IEQ^{(i)} \cap Q_j^{(i)}$ whose relative order in $Q_j^{(i)}$ is the opposite of that in $IEQ^{(i)}$. n is the union size.

2.1.6 Spearman's Rank Correlation Coefficient

Spearman's Rank Correlation Coefficient (ρ):

$$Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \frac{\sum_n \left(r_n^{IEQ^{(i)}} - r^{\bar{IEQ^{(i)}}}\right)\left(r_n^{Q_j^{(i)}} - r^{\bar{Q_j^{(i)}}}\right)}{\sqrt{\sum_n \left(r_n^{IEQ^{(i)}} - r^{\bar{IEQ^{(i)}}}\right)^2 \left(r_n^{Q_j^{(i)}} - r^{\bar{Q_j^{(i)}}}\right)^2}}$$

where $r_n^{IEQ^{(i)}}$ and $r_n^{Q_j^{(i)}}$ are the ranks of n^{th} term belonging to $IEQ^{(i)} \cap Q_j^{(i)}$ in $IEQ^{(i)}$ and $Q_j^{(i)}$. $r^{\bar{IEQ^{(i)}}}$ and $r^{\bar{Q_j^{(i)}}}$ are the means of ranks of terms of $IEQ^{(i)}$ and $Q_j^{(i)}$. Here only terms belonging to both $IEQ^{(i)}$ and $Q_j^{(i)}$ are considered.

2.1.7 Result

	Pearson	Kendall	Spearman
Jaccard index	0.558512	0.439212	0.577354
Inner Product L_1 Norm	-0.321893	-0.274142	-0.352637
Inner Product L_2 Norm	0.306361	0.328803	0.412454
Inner Product L_∞ Norm	0.429213	0.352161	0.444382
Inner Product Unnormalized	0.419395	0.375006	0.485278
NDCG	0.27668	0.35067	0.45515
Pearson	0.028628	-0.025947	-0.044691
Kendall	-0.132430	-0.10328	-0.144502
Spearman	-0.121821	-0.105900	-0.150395

Table 2.1: Existing Correlation Metrics for 225-element-set

	Pearson	Kendall	Spearman
Jaccard index	0.736682	0.604199	0.739018
Inner Product L_1 Norm	-0.538969	-0.419434	-0.524876
Inner Product L_2 Norm	0.639795	0.498164	0.640914
Inner Product L_∞ Norm	0.677685	0.544253	0.679660
Inner Product Unnormalized	0.651261	0.531109	0.662361
NDCG	0.49919	0.47927	0.60287
Pearson	-0.037164	-0.026907	-0.051484
Kendall	-0.191254	-0.141127	-0.192506
Spearman	-0.198278	-0.154138	-0.214458

Table 2.2: Existing Correlation Metrics for 45-element-set of DFR Variant

	Pearson	Kendall	Spearman
Jaccard index	0.747630	0.621196	0.763451
Inner Product L_1 Norm	-0.533632	-0.436281	-0.543391
Inner Product L_2 Norm	0.658796	0.523195	0.669392
Inner Product L_∞ Norm	0.635724	0.515586	0.649033
Inner Product Unnormalized	0.640888	0.513360	0.646433
NDCG	0.51957	0.50958	0.64167
Pearson	-0.060492	-0.032890	-0.063408
Kendall	-0.219706	-0.175159	-0.233796
Spearman	-0.212889	-0.170887	-0.230653

Table 2.3: Existing Correlation Metrics for 45-element-set of KLD Variant

	Pearson	Kendall	Spearman
Jaccard index	0.653996	0.507467	0.631757
Inner Product L_1 Norm	−0.544778	−0.415812	−0.498609
Inner Product L_2 Norm	0.051115	0.022250	0.045036
Inner Product L_∞ Norm	0.618788	0.520190	0.616143
Inner Product Unnormalized	0.682740	0.578685	0.665302
NDCG	0.33268	0.42062	0.49040
Pearson	−0.035513	−0.026497	−0.045504
Kendall	−0.096548	−0.099416	−0.126994
Spearman	−0.080004	−0.105960	−0.134169

Table 2.4: Existing Correlation Metrics for 45-element-set of LCA Variant

	Pearson	Kendall	Spearman
Jaccard index	0.692599	0.562692	0.671922
Inner Product L_1 Norm	−0.421535	−0.318506	−0.404660
Inner Product L_2 Norm	0.446192	0.384541	0.479864
Inner Product L_∞ Norm	0.599938	0.480292	0.585377
Inner Product Unnormalized	0.606207	0.481689	0.586575
NDCG	0.51218	0.45220	0.55906
Pearson	−0.040316	−0.047056	−0.076093
Kendall	−0.083330	−0.073668	−0.107263
Spearman	−0.079093	−0.075885	−0.112744

Table 2.5: Existing Correlation Metrics for 45-element-set of LCANEW Variant

	Pearson	Kendall	Spearman
Jaccard index	0.7826450	0.6297879	0.7687967
Inner Product L_1 Norm	−0.8475117	−0.7326186	−0.8586733
Inner Product L_2 Norm	0.6598335	0.5547841	0.6909573
Inner Product L_∞ Norm	0.7181386	0.6019089	0.7397788
Inner Product Unnormalized	0.4936310	0.4401082	0.5588896
NDCG	0.51895	0.53755	0.66888
Pearson	−0.0048715	−0.0270240	−0.0075296
Kendall	−0.1686076	−0.0871832	−0.1312030
Spearman	−0.1639409	−0.0820235	−0.1295021

Table 2.6: Existing Correlation Metrics for 45-element-set of RBLM Variant

2.1.8 Analysis

1. Jaccard Index: In spite of being the simplest among the Sim_{Query} variants, this gives the best value for our 225-element set consisting of all 5 variants. But its drawback comes from the fact that it ignores term-weights. It never checks the quality of the terms selected by the candidate expanded queries from the ideal expanded query. It just counts the number of such terms.

2. Inner Product Similarity:

   (a) It is quite counter-intuitive that L_1 Norm performs so poorly for all variants. One probable reason can be the existence of dominance in term-weights of candidate terms. But how these dominant terms affect the L_1 Norm metric is not clear.

   (b) The average of L_2 Norm has basically dropped for its poor performance in case of LCA variations.

   (c) L_∞ Norm and Unnormalized versions are very close except for RBLM variation where L_∞ Norm is much better.

   (d) The main disadvantage of the Unnormalized version is that it is highly susceptible to scaling. Suppose $IEQ^{(i)}$ is an ideal query and $Q_1^{(i)}$,$Q_2^{(i)}$ are two candidate expanded queries.

   Let

   $$IEQ^{(i)} = T_1^p + T_2^q + T_3^r,$$

   $$Q_1^{(i)} = T_1^a + T_3^b + T_4^c$$

   and

   $$Q_2^{(i)} = T_1^{n*a} + T_3^{n*b} + T_4^{n*c}$$

   where $IEQ^{(i)}, Q_1^{(i)}$ and $Q_2^{(i)}$ are represented in vector notation. T_1, T_2, T_3, T_4 are query-terms; p, q, r, a, b, c are term-weights and n is a constant.

   Then the performance of $Q_1^{(i)}$ and $Q_2^{(i)}$ in terms of MAP will be same, but Inner Product Similarity (Unnormalized) of $Q_2^{(i)}$ (i.e. $n * (a * p + b * r)$) will be n times more than that of $Q_1^{(i)}$. This is certainly not desirable and it is also highly counter-intuitive that in spite of possessing such a drawback this metric performs equally well as other normalized versions.

3. : Pearson, Kendall and Spearman's Correlation Coefficients: This three metrics consistently perform poorly. To understand the reason behind this, we performed the following experiment for each of them:

   (a) Find those queries having the least values (<-0.7) for the metric.

   (b) For each such query find those variations which caused that poor value. We selected two types of variations : firstly those having high similarity (top 20%) but low MAP (bottom 20%), secondly those having high MAP (top 20%) but low similarity (bottom 20%).

   (c) Analyse the no. of intersection terms, their ranks, weights etc. for each such variation.

   After careful analysis, we found that these three metrics concentrate only on the order of the terms selected from the ideal query by the candidate query. So, they totally disregard the count or goodness of the intersecting terms. This is the main factor behind their poor performance.

   Consider, for example, two variants $Q_1^{(i)}$ and $Q_2^{(i)}$. Suppose $|IEQ^{(i)} \cap Q_1^{(i)}| = 3$ and $|IEQ^{(i)} \cap Q_2^{(i)}| = 10$. If the terms in $Q_1^{(i)}$ are ranked in the same order as in $IEQ^{(i)}$, $Sim_{Query}(IEQ^{(i)}, Q_1^{(i)})$ will be very high, but the MAP for this query will be low, especially if the 3 common terms are relatively less important in $IEQ^{(i)}$. On the other hand, If the terms in $Q_2^{(i)}$ are not ranked in the same order as in $IEQ^{(i)}$, $Sim_{Query}(IEQ^{(i)}, Q_2^{(i)})$ will be very low, but the MAP for this query will be high, especially if the 10 common terms are relatively more important in $IEQ^{(i)}$.

   We calculated that on an average there exists 5 no. of intersecting terms for high-similarity-low-MAP cases and 16 no. of intersecting terms for low-similarity-high-MAP cases. So, these 3 metrics are not quite suitable to be used as Sim_{Query}.

2.2 New Metrics

2.2.1 Modified Jaccard Index

General Jaccard index counts the no. of intersecting terms but it does not consider their goodness. Let two n-terms-long candidate expanded queries be $Q_1^{(i)}$ and $Q_2^{(i)}$. Suppose both of them have 5 terms in common with the corresponding ideal expanded query $IEQ^{(i)}$. $Q_1^{(i)}$ contains the top 5 terms of $IEQ^{(i)}$ (in terms of term-weights) and $Q_2^{(i)}$ contains the bottom 5 terms of $IEQ^{(i)}$. Jaccard Index values for $Q_1^{(i)}$ and $Q_2^{(i)}$ will be same but $Q_1^{(i)}$ is expected to perform much better than $Q_2^{(i)}$.

So, to address this drawback Jaccard Index is modified in the following two ways:

2.2.1.1 Modification 1

Here the sum of the term-weights of the intersecting terms is calculated where term-weights are taken from $IEQ^{(i)}$.

Jaccard Index Modification 1: $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \sum_{t \in IEQ^{(i)} \cap Q_j^{(i)}} W_t^{IEQ^{(i)}}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are the ideal and candidate queries respectively, represented in set notation, and $W_t^{IEQ^{(i)}}$ is the weight of the term t in $IEQ^{(i)}$.

2.2.1.2 Modification 2

Here the sum of the reciprocals of the term-ranks of the intersecting terms is calculated where term-ranks are taken from $IEQ^{(i)}$.

Jaccard Index Modification 2: $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \sum_{t \in IEQ^{(i)} \cap Q_j^{(i)}} \frac{1}{R_t^{IEQ^{(i)}}}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are the ideal and candidate queries respectively, represented in set notation, and $R_t^{IEQ^{(i)}}$ is the rank of the term t in $IEQ^{(i)}$.

2.2.2 Modified Kendall's Rank Correlation Coefficient

In general, Kendall Rank Correlation Coefficient (τ) works well for measuring similarity between two lists having the same set of terms. But it does not handle those cases where there exists such terms in both ideal and candidate expanded queries, which do not belong to intersection of them. This is very common in our case. If we choose a pair of terms randomly from the union of ideal and candidate queries, anyone or both of the terms may not belong to their intersection. Now, the question is whether to treat them as concordant pairs or discordant pairs. So, we have modified Kendall Tau as required for our purpose and tried to decide the concordance/discordance of the pairs based on some conditions.

2.2.2.1 Modification 1

1. Let, $U = IEQ^{(i)} \cup Q_j^{(i)}$.
2. Now, let us choose two terms randomly from U. Let the terms be T_1 and T_2.
3. (a) Case 1: T_1 belongs to $IEQ^{(i)}$ and it does not belong to $Q_j^{(i)}$
    - i. T_2 belongs to $IEQ^{(i)}$ and it does not belong to $Q_j^{(i)}$: In this case we cannot say anything about concordance or discordance as both the terms belong only to the ideal query. So, to be on the safer side we count it as discordance. This in fact is a very important case because occurs hit about 75% times.
    - ii. T_2 belongs to both $IEQ^{(i)}$ and $Q_j^{(i)}$: As T_1 does not belong to $Q_j^{(i)}$ and T_2 belongs to $Q_j^{(i)}$, so we can safely assume that rank of T_2 is higher than rank of T_1 in $Q_j^{(i)}$. So, if rank of T_2 is also higher than rank of T_1 in $IEQ^{(i)}$, then it's a concordance otherwise it's a discordance.
    - iii. T_2 belongs to $Q_j^{(i)}$ and it does not belong to $IEQ^{(i)}$: As T_1 does not belong to $Q_j^{(i)}$ and T_2 belongs to $Q_j^{(i)}$, so we can safely assume that rank of T_2 is higher than rank of T_1 in $Q_j^{(i)}$. Similarly, as T_2 does not belong to $IEQ^{(i)}$ and T_1 belongs to $IEQ^{(i)}$, we can assume that rank of T_1 is higher than rank of T_2 in $IEQ^{(i)}$. So, it's undoubtedly a discordance.

   (b) Case 2: T_1 belongs to $IEQ^{(i)}$ and $Q_j^{(i)}$
    - i. T_2 belongs to $IEQ^{(i)}$ and it does not belong to $Q_j^{(i)}$: As T_2 does not belong to $Q_j^{(i)}$ and T_1 belongs to $Q_j^{(i)}$, so in $Q_j^{(i)}$ rank of T_1 is higher than rank of T_2. If in $IEQ^{(i)}$ also, rank of T_1 is greater than rank of T_2 then the pair is concordant otherwise it is discordant.

ii. T_2 belongs to both $IEQ^{(i)}$ and $Q_j^{(i)}$: This is the case of normal Kendall τ. So, the pair is concordant when rank of T_1 is greater than rank of T_2 in both $IEQ^{(i)}$ and $Q_j^{(i)}$ OR when rank of T_1 is less than/equal to rank of T_2 in both $IEQ^{(i)}$ and $Q_j^{(i)}$. Otherwise it is discordant.

iii. T_2 belongs to $Q_j^{(i)}$ and it does not belong to $IEQ^{(i)}$: As T_2 does not belong to $IEQ^{(i)}$, we can assume that rank of T_1 in $IEQ^{(i)}$ is higher than rank of T_2 in $IEQ^{(i)}$. If for $Q_j^{(i)}$ also rank of T_1 is higher than rank of T_2, then the pair is concordant otherwise it is discordant.

(c) Case 3: T_1 belongs to $Q_j^{(i)}$ but it does not belong to $IEQ^{(i)}$

i. T_2 belongs to $IEQ^{(i)}$ and it does not belong to $Q_j^{(i)}$: As T_2 does not belong to $Q_j^{(i)}$ and T_1 belongs to $Q_j^{(i)}$, so in $Q_j^{(i)}$ rank of T_1 is higher than rank of T_2. On the other hand, as T_1 does not belong to $IEQ^{(i)}$ and T_2 belongs to $IEQ^{(i)}$, so rank of T_2 in $IEQ^{(i)}$ is higher than the rank of T_1 in $IEQ^{(i)}$. So, this case is definitely discordant.

ii. T_2 belongs to both $IEQ^{(i)}$ and $Q_j^{(i)}$: T_1 does not belong to $IEQ^{(i)}$ but T_2 belongs to $IEQ^{(i)}$. This means that rank of T_2 is higher than rank of T_1 in $IEQ^{(i)}$. If for $Q_j^{(i)}$ also rank of T_2 is higher than rank of T_1 then only the pair is concordant.

iii. T_2 belongs to $Q_j^{(i)}$ and it does not belong to $IEQ^{(i)}$: None of T_1 and T_2 belong to $IEQ^{(i)}$. So, we can not say anything about the concordance/discordance of this pair. For safe calculations, we have assumed this as a case of discordance.

4. Now finally calculate the modified Kendall correlation coefficient in the same way as the original.

$$Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \frac{\text{no of concordant pairs} - \text{no of discordant pairs}}{\frac{1}{2}|U|(|U|-1)}$$

2.2.2.2 Modification 2

This is a relatively lenient version of the above variation. Here in case of discordant pairs, the difference between the weights of the discordant terms is taken into account. If that is less than a threshold, then the pair is considered as a concordant pair.

T_1	T_2	Treated as Concordance if
In $IEQ^{(i)}$, not in $Q_j^{(i)}$	In $IEQ^{(i)}$, not in $Q_j^{(i)}$	Never
In $IEQ^{(i)}$, not in $Q_j^{(i)}$	In $IEQ^{(i)}$ and in $Q_j^{(i)}$	In $IEQ^{(i)}$, rank(T_1) $<$ rank(T_2)
In $IEQ^{(i)}$, not in $Q_j^{(i)}$	In $Q_j^{(i)}$, not in $IEQ^{(i)}$	Never
In $IEQ^{(i)}$ and in $Q_j^{(i)}$	In $IEQ^{(i)}$, not in $Q_j^{(i)}$	In $IEQ^{(i)}$, rank(T_1) $>$ rank(T_2)
In $IEQ^{(i)}$ and in $Q_j^{(i)}$	In $IEQ^{(i)}$ and in $Q_j^{(i)}$	In both $IEQ^{(i)}$ and $Q_j^{(i)}$, either rank(T_1) $>$ rank(T_2) or rank(T_1) $<=$ rank(T_2)
In $IEQ^{(i)}$ and in $Q_j^{(i)}$	In $Q_j^{(i)}$, not in $IEQ^{(i)}$	In $Q_j^{(i)}$, rank(T_1) $>$ rank(T_2)
In $Q_j^{(i)}$, not in $IEQ^{(i)}$	In $IEQ^{(i)}$, not in $Q_j^{(i)}$	Never
In $Q_j^{(i)}$, not in $IEQ^{(i)}$	In $IEQ^{(i)}$ and in $Q_j^{(i)}$	In $Q_j^{(i)}$, rank(T_1) $<$ rank(T_2)
In $Q_j^{(i)}$, not in $IEQ^{(i)}$	In $Q_j^{(i)}$, not in $IEQ^{(i)}$	Never

Table 2.7: Modified Kendall Rank Correlation - Modification 1

Selection of Threshold:

As the range of weights in ideal ($IEQ^{(i)}$) and candidate queries ($Q_j^{(i)}$) are different, we need two thresholds Θ_1 and Θ_2, one for the ideal query and one for the candidate query.

Θ_1: This can be taken as a percentage of average weight or maximum weight of $IEQ^{(i)}$.

Θ_2: This can be taken as a percentage of average weight or maximum weight of $Q_j^{(i)}$.

2.2.2.3 Modification 3

In the first modification, we simply counted the no. of concordant and discordant pairs. But here each concordance and discordance is assigned a weight [9]. That weight is basically influenced by two factors:

1. Element weights: As the higher weighted terms are considered to be more important, concordance and discordance among the higher weighted terms should be given more importance than concordance and discordance among the lower weighted terms. We, therefore, use the ideal weights of the terms belonging to the selected pair, as a multiplying factor.
2. Element similarities: While element weights address the question of swaps occurring near the beginning or an end of a permutation, many times the importance of

the swap crucially depends on the similarity of the elements being swapped. In an extreme case, swapping two identical elements should result in no change to the metric, whereas swapping two radically different elements should result in a large effect, even if the weights are small. So, to take element similarities in account, the weight difference between the pair of terms (weights taken from $IEQ^{(i)}$) may also be used as a multiplying factor.

Here, a single score is used which is increased for concordant pairs and decreased for discordant pairs in the following way:

Let T_1 and T_2 be the terms chosen randomly from U where U is the union of terms belonging to ideal expanded query $IEQ^{(i)}$ and candidate expanded query $Q_j^{(i)}$.

For concordance,

score = score + (weight of T_1 from $IEQ^{(i)}$)*(weight of T_2 from $IEQ^{(i)}$)*(absolute difference of weights of T_1 and T_2)

For discordance,

score = score - (weight of T_1 from $IEQ^{(i)}$)*(weight of T_2 from $IEQ^{(i)}$)*(absolute difference of weights of T_1 and T_2)

If anyone of T_1 and T_2 does not belong to $IEQ^{(i)}$, its weight is replaced by the average term-weight of $IEQ^{(i)}$

Finally for normalization, the score is divided by $|U|$*(($|U|$-1)/2) where $|U|$ is the union size.

2.2.3 Modified NDCG

2.2.3.1 Modification 1

Previously, in general NDCG the gain function was exponential. Here a linear gain function is used. Again, to reduce the discount, the ranks of the intersecting terms among only the intersecting terms are used whereas in the original function ranks of the intersecting terms among all the terms of the candidate query were used.

Modified NDCG (Modification 1): $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \sum_{t \in IEQ^{(i)} \cap Q_j^{(i)}} \frac{W_t^{IEQ^{(i)}}}{log(1+\text{Rnew}_t^{Q_j^{(i)}})}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are represented in set notation, $W_t^{IEQ^{(i)}}$ is the weight of term t in $IEQ^{(i)}$ and $\text{Rnew}_t^{Q_j^{(i)}}$ is the rank of the term t in $IEQ^{(i)} \cap Q_j^{(i)}$ based on weights from $Q_j^{(i)}$.

2.2.3.2 Modification 2

This applies just a little modification on the discounting function of the previous one. In the previous modification we used logarithmic discounting function. Here that is modified to $\frac{K}{K+x}$ form.

Modified NDCG (Modification 2): $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)}) = \sum_{t \in IEQ^{(i)} \cap Q_j^{(i)}} \frac{W_t^{IEQ^{(i)}} * K}{K+\text{Rnew}_t^{Q_j^{(i)}}}$

where $IEQ^{(i)}$ and $Q_j^{(i)}$ are represented in set notation, $W_t^{IEQ^{(i)}}$ is the weight of term t in $IEQ^{(i)}$ and $\text{Rnew}_t^{Q_j^{(i)}}$ is the rank of the term t in $IEQ^{(i)} \cap Q_j^{(i)}$ based on weights from $Q_j^{(i)}$. K is a constant. Here we have used K=100.

2.2.4 Result

	Pearson	Kendall	Spearman
Modified Jaccard Index Modification 1	0.609345	0.477624	0.622117
Modified Jaccard Index Modification 2	0.568972	0.467555	0.609813
Modified Kendall rank Modification 1	0.58400	0.45014	0.59293
Modified Kendall rank Modification 2(5% Tolerance)	0.58110	0.44970	0.59334
Modified Kendall rank Modification 3	0.49527	0.38649	0.51929
Modified NDCG Modification 1	0.59481	0.46831	0.61108
Modified NDCG Modification 2	0.61155	0.47609	0.62229

Table 2.8: Modified Correlation Metrics for 225-element-set

	Pearson	Kendall	Spearman
Modified Jaccard Index Modification 1	0.738764	0.611565	0.748209
Modified Jaccard Index Modification 2	0.686228	0.598325	0.733332
Modified Kendall rank Modification 1	0.74553	0.60069	0.74512
Modified Kendall rank Modification 2(5% Tolerance)	0.74280	0.59993	0.74273
Modified Kendall rank Modification 3	0.56782	0.46358	0.58647
Modified NDCG Modification 1	0.74689	0.61196	0.75308
Modified NDCG Modification 2	0.74039	0.60562	0.74559

Table 2.9: Modified Correlation Metrics for 45-element-set of DFR Variant

	Pearson	Kendall	Spearman
Modified Jaccard Index Modification 1	0.752889	0.633273	0.773281
Modified Jaccard Index Modification 2	0.708425	0.620701	0.759729
Modified Kendall rank Modification 1	0.75702	0.62084	0.77041
Modified Kendall rank Modification 2(5% Tolerance)	0.75439	0.61835	0.76784
Modified Kendall rank Modification 3	0.59549	0.50018	0.62549
Modified NDCG Modification 1	0.76727	0.64110	0.78546
Modified NDCG Modification 2	0.75496	0.63112	0.77367

Table 2.10: Modified Correlation Metrics for 45-element-set of KLD Variant

	Pearson	Kendall	Spearman
Modified Jaccard Index Modification 1	0.642716	0.555810	0.649509
Modified Jaccard Index Modification 2	0.598220	0.548810	0.640968
Modified Kendall rank Modification 1	0.65892	0.51799	0.63880
Modified Kendall rank Modification 2(5% Tolerance)	0.65786	0.51606	0.63771
Modified Kendall rank Modification 3	0.48432	0.42281	0.50300
Modified NDCG Modification 1	0.64712	0.54432	0.64274
Modified NDCG Modification 2	0.64555	0.55082	0.64805

Table 2.11: Modified Correlation Metrics for 45-element-set of LCA Variant

	Pearson	Kendall	Spearman
Modified Jaccard Index Modification 1	0.688367	0.571537	0.694557
Modified Jaccard Index Modification 2	0.669122	0.562692	0.671922
Modified Kendall rank Modification 1	0.69235	0.56292	0.67719
Modified Kendall rank Modification 2(5% Tolerance)	0.69093	0.56290	0.67460
Modified Kendall rank Modification 3	0.55696	0.45170	0.54975
Modified NDCG Modification 1	0.65974	0.55427	0.65667
Modified NDCG Modification 2	0.68946	0.56957	0.67503

Table 2.12: Modified Correlation Metrics for 45-element-set of LCANEW Variant

	Pearson	Kendall	Spearman
Modified Jaccard Index Modification 1	0.7986536	0.7303869	0.8403749
Modified Jaccard Index Modification 2	0.7662713	0.7334866	0.8425134
Modified Kendall rank Modification 1	0.77774	0.61812	0.76066
Modified Kendall rank Modification 2(5% Tolerance)	0.78371	0.61525	0.76409
Modified Kendall rank Modification 3	0.69365	0.61788	0.72092
Modified NDCG Modification 1	0.79194	0.70700	0.82787
Modified NDCG Modification 2	0.79908	0.70844	0.83154

Table 2.13: Modified Correlation Metrics for 45-element-set of RBLM Variant

2.2.5 Analysis

1. Modified Jaccard Index Modification 1: This so far gives the best results. Except LCA and LCANEW variations it performs quite well (above 0.74) for the other three groups of variations (especially for RBLM). But it still has a drawback that it does not take care of how a term selected from ideal expanded query is weighted in candidate expanded queries.

2. Modified Kendall versions 1 and 2 performs quite well. But the mostly hit case (around 80%) here is when both of the terms of the chosen pair belong only to IEQ or only to candidate expanded query, and this case has probably not been well handled here. In order to be conservative, we just assumed such pairs to be discordant, but that assumption may not be right.

2.3 Poor Queries

We checked the overlap among the 12 queries with the lowest Sim_{ranks} scores for Jaccard Index, Modified Jaccard Index Modification 1, Modified Kendall rank modification 1 and Modified NDCG Modification 1. We found as many as 10 queries in common across the 4 similarity measures. These 10 queries are 401, 409, 419, 429, 430, 432, 433, 440, 442 and 448.

2.3.1 Reasons Behind Poor Performance

For some of these 10 poor queries, some probable reasons for their poor performance are given below.

1. For 433, all candidate expanded queries have MAP=0. So, our hypothesis does not apply in this case.

2. For 401, 432, 442 and 448 the range and standard deviations of the MAP values achieved by the candidate queries are very low. So, for these queries, the correlation among the $Sim_{Query}(IEQ^{(i)}, Q_j^{(i)})$ values and variant MAPs is in any case not expected to be high.

Query	Mean MAP	Range of MAP	Standard Deviation
401	0.01471	0.0391	0.00723
432	0.00028	0.0017	0.00026
442	0.00621	0.0336	0.00805
448	0.00504	0.0249	0.00381

TABLE 2.14: Mean, Range and Standard Deviation of MAP for some poor queries

3. For 429 and 430, the no. of relevant documents is very low (just 11 and 6 respectively).

2.3.2 Result Without Poor Queries

Results of some of the best performing metrics are given below without counting the 10 poor queries:

	Jaccard Index	Modified Jaccard 1	Modified Kendall 1	Modified NDCG 1
overall	0.67754	0.71609	0.68811	0.71584
DFR	0.83194	0.84399	0.84185	0.84881
KLD	0.83351	0.84528	0.84454	0.85434
LCA	0.75195	0.77666	0.75512	0.76543
LCANEW	0.79318	0.78932	0.79979	0.76276
RBLM	0.80612	0.88157	0.80833	0.87254

TABLE 2.15: Performance without poor queries

Chapter 3

Conclusion

3.1 Summary of Findings

1. Metrics for measuring overlap/correlation between IEQ and candidate expanded queries:

   Among the standard correlation metrics Jaccard index performs the best. Among the new metrics, except modification 3 of Kendall (relatively poor), all perform almost equally well. These modified metrics perform relatively better than Jaccard index.

2. On variant basis, maximum correlation values are given below:

Variation	Maximum Correlation Value	Achieved by Metric
DFR	0.75300	NDCG Modification 1-Spearman
KLD	0.78546	NDCG Modification 1-Spearman
LCA	0.68274	Inner Product Unnormalized-Pearson
LCANEW	0.69456	Jaccard index Modification 1-Spearman
RBLM	0.84251	Jaccard Index Modification 2-Spearman

Table 3.1: Best Correlation Metrics for each variation

3. For the mixture of all these 5 variations we are getting maximum 0.62229 similarity (NDCG Modification 2-Spearman) which is substantially less than the average of all 5 variations (0.75). The most probable reason for this is as follows. If we have two pairs of highly correlated lists, it does not guarantee high correlation after mixing them up.

For example, suppose the first pair of lists is:

(1,5,7) and (0.3,0.5,0.8) (this pair of lists has Spearman correlation 1)

and suppose the second pair of lists is:

(2,6,8) and (0.25,0.4,0.6) (this pair of lists also has Spearman correlation 1)

Now, the mixture of this two pair of lists will be

(1,2,5,6,7,8) and (0.3,0.25,0.5,0.4,0.8,0.6)

This mixture pair of lists has much less correlation (0.8) than both of its constituents.

Apart from this, lack of proper scaling/normalization while checking the overall correlation, can be another probable reason.

4. While finding correlation between IEQ and candidate expanded queries, Pearson, Kendall and Spearman's correlation coefficients are found to be quite poor (in the range 0-(-0.2)). The reason found behind this is their tendancy to give the order of the lists more importance than the actual list data. The detailed explanation is given in Section 2.1.8.

5. The reason behind poor performance of Inner Product L_1 norm is not yet known. Inner Product Unnormalized metric was expected to perform worse than all three normalized versions of Inner Product similarity. But this did not happen; instead Inner Product Unnormalized metric sometimes gave better results among all variations of Inner ProductSimilarity metric.

3.2 Future Work

1. It is not at all clear why Inner Product L_1 norm performs so badly.

2. The results for LCA variations are quite confusing. For this variation, Inner Product unnormalized metric gives the best values which is not at all expected. Inner Product L_2 Norm is almost 0 here whereas it gives 0.5-0.65 values for other variations. The metrics like Jaccard modifications, Kendall modifications, NDCG modifications could not perform that well for LCA variations. So, the reasons behind these confusing results are to be found out by carefully observing the LCA expanded queries.

3. We have found a reasonably good overlap among the queries performing poorly for Jaccard modifications, Kendall modifications and NDCG modifications. Some probable reasons behind their poor performance have been discussed in Section 2.3.1. The complete characterisation is yet to be done.

Bibliography

[1] Text retrieval conference. http://trec.nist.gov/.

[2] Wikipedia link on information retrieval. http://en.wikipedia.org/wiki/Information_retrieval.

[3] Wikipedia link on tf-idf. http://en.wikipedia.org/wiki/Tf%E2%80%93idf.

[4] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389, 2002.

[5] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1, 2012.

[6] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971, 1987.

[7] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[8] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.

[9] Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 571–580, New York, NY, USA, 2010. ACM.

[10] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and*

development in information retrieval, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.

[11] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval.* Cambridge University Press, 2008.

[12] Snehasish Mukherjee. Relative query performance prediction using ideal expanded query. Master's thesis, Indian Statistical Institute, 2012.

[13] Dipasree Pal, Mandar Mitra, and Kalyankumar Datta. Query expansion using term distribution and term association. *CoRR*, abs/1303.0667, 2013.

[14] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[15] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. A theoretical analysis of ndcg type ranking measures. *CoRR*, abs/1304.6480, 2013.

[16] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.