IMPROVEMENT IN PROBABILISTIC INFORMATION RETRIEVAL MODEL - REWARDING TERMS WITH HIGH RELATIVE TERM FREQUENCY

RUNJIE ZHU

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

Master of Arts

GRADUATE PROGRAM IN INFORMATION SYSTEMS AND TECHNOLOGY

York University
Toronto, Canada

JUNE 2016

Abstract

Information retrieval has been one of the most popular research fields in the past decades. While term weighting schemes are the central to the information retrieval systems. The basic concept of the information retrieval system is that when a user sends out a query, the system would try to generate a list of related documents ranked in order, according to their degree of relevance. Most of the present information retrieval systems assign numeric scores by weighting functions to certain documents, and put them in rank based on the scores. Same as other information retrieval models, in the context of probabilistic model, the main factors affecting the computation of a term's weight include (i) a within document frequency of the term, (ii) a document frequency of the term in the collection, and (iii) the length of the document where the term sits in. This thesis emphasize on the result of the integration of relative term frequency weighting and the term frequency normalization based on document length, and its application to the classic probabilistic weighting function of BM25.

To elaborate it in more details, in this thesis, I propose the relative term frequency to be integrated into traditional probabilistic models, in other

words, I introduce a set of three influence functions with the application of relative term frequency to model and enhance the performance of the fundamental probabilistic weighting function, BM25. The study aims to exploit the properties of the combination of relative term frequency and BM25. The extensive experiments and analyses conducted in the thesis are based on six of the TREC official datasets, and the results presented have shown a significant improvement in the retrieval effectiveness. The information retrieval system adopted is built on the Okapi Basic Search System (BSS), which offers a reliable and effective packaged framework to exercise the experiments, and to yield an end-to-end retrieval workflow.

Acknowledgement

First of all, I would like to express my sincere gratitude to Professor Jimmy Xiangji Huang for his continuous guidance, motivation, leadership, enthusiasm and thought provoking ideas. During my master study with Prof. Jimmy Xiangji Huang, I was introduced to information retrieval and started to do research beyond this area. With his guidance and patience, I was able to change my specialization smoothly from an undergraduate Political Science and Economics double major degree to a Master in Information Systems and Technology. Professor Jimmy Xiangji Huang has always kept a strong confidence in me, and has provided me a lot of opportunities to conferences, seminars and introduced many famous professors and scholars to me in this field. Moreover, he was never too busy to sit down with me and went through some details with me during the research time. I feel so grateful to have Professor Huang as my Master supervisor and mentor.

Besides Professor Huang, I would like to send my appreciation to the rest of my thesis committee as well: Professor George J. Gerogopoulos from the Economics Department and Professor Augustine Wong, for their advice, patience, continuous guidance, and insightful comments.

Last but not least, I would like to thank my family. My best father Xianlin Zhu and best mother Li Zhang have always been my strongest support, and take care of me from all well round aspects. When I met difficulties in my research and study, I always remember what my father told me, “just walk ahead, and don’t be scared, we will always be there for you”. It is because of their love and support; I can stay concentrated on academic work and write my thesis with a peaceful mind.

Table of Contents

Abbreviations

Here is a list of abbreviations mentioned and discussed throughout this thesis:

Okapi Basic Search System (BSS): The Okapi BSS is an information retrieval ranking function used by the search engines to rank matching documents according to their relevance to the user's given search query. It helps to index text documents and also assigns weighting scores to terms in documents with weighting functions.

Best Match 25 (BM25): The BM25 is a classic weighting function in probabilistic information retrieval approach. It was initially proposed by the famous IR field scientists and researchers Stephen E. Robertson and Karen Sparck Jones.

Term frequency (tf): a numerical statistic concept, applied in term weighting schemes of text mining, to calculate how important a term is to the selected document. A term frequency simply suggests that the weight of a term occurring in a document is simply proportional to the term frequency.

Inverse document frequency(idf): idf implies that the specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.

tf-idf: the value will increase proportionally to the number of times a term appears in the document, but the effect will be balanced out by the frequency of the term appearing in the document collections.

Precision: $\frac{the\ total\ number\ of\ documents\ retreived\ that\ are\ relevant\ to\ the\ given\ query}{the\ total\ number\ of\ retrieved\ documents}$, the proportion of how many documents retrieved are evaluated as relevant to the user's given query.

Precision@N or P@N: Only focus the evaluation on the top N documents instead of reading the precision of the whole document collection retrieved. Compared to traditional precision, it is more practical and effective, since in most cases, users tend to be more interested in obtaining the most relevant documents on top of their list.

Mean Average Precision (MAP): the average precision score is calculated each time for the single query after one relevant document is retrieved. Then the MAP will take the average precision scores and calculate the mean of these values.

Chapter 1 Introduction

1.1 Background

In the age of information today, can you still imagine living in a world without internet? It is almost impossible. It has been quietly and deeply infiltrated into our everyday life. The internet in the past decade has gone through an explosive growth and changes. People stick to the internet day and night to fulfill their various needs of information. Thus, the accuracy and efficiency of information extraction to meet our needs become central to the study of information retrieval systems.

Information Retrieval (IR) aims to find relevant information resources to a query from a collection of information resources, where a query is either a short or long statement with series of keywords of user's information need, and the retrieved result is a list of ranked documents of the data collection. Traditionally, queries are converted to the query representations while documents are returned into the indexed document representations in order to increase the effectiveness and efficiency of the retrieval system. An

automated information retrieval system takes the query as input, and outputs a list of ranked documents ordered by degree of relevancy.

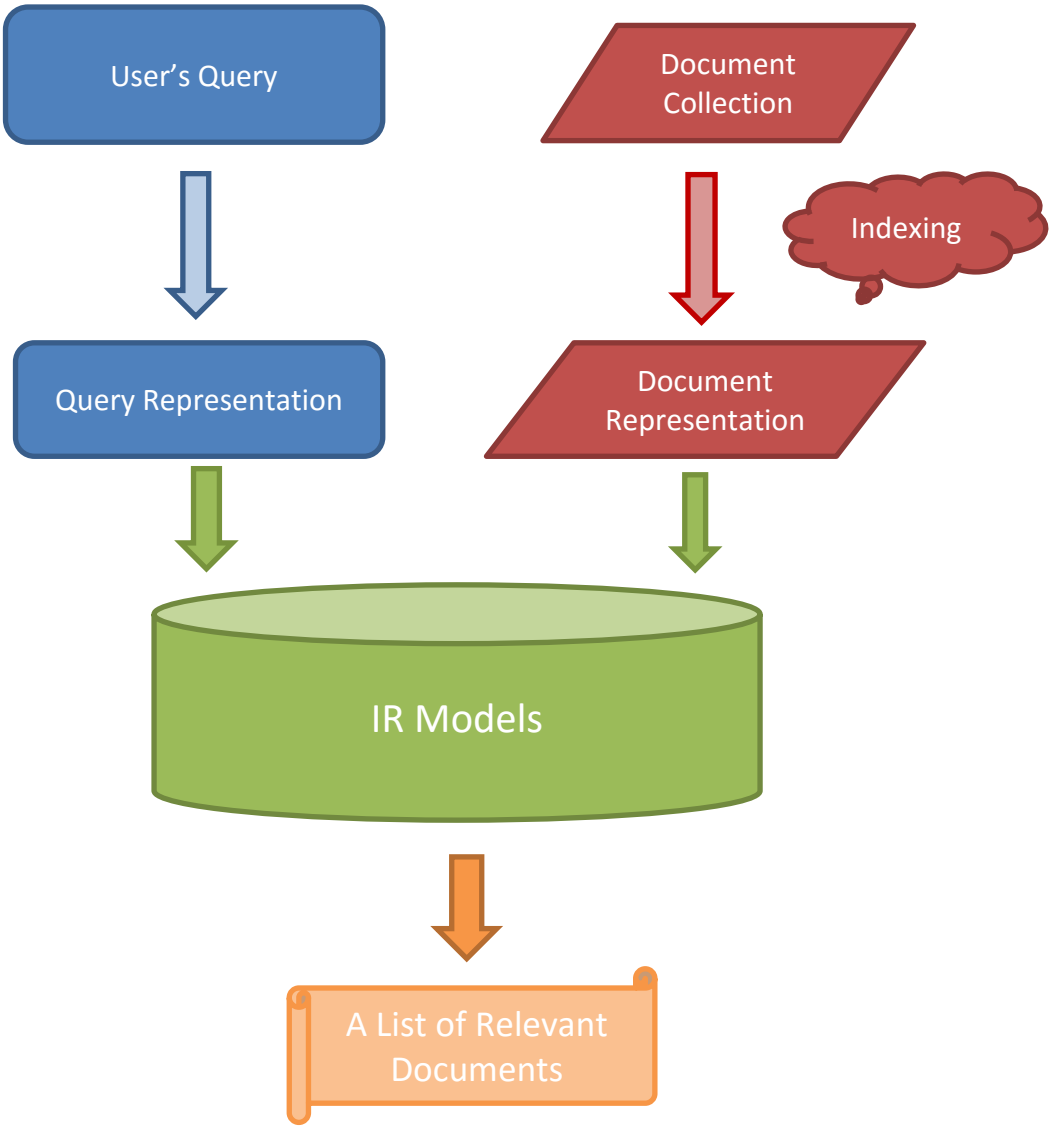

Figure 1. Basic IR System

Figure 1. is the structure of a basic IR system, where the traditional IR model processes to match the query representation with the document representation. The system starts with the users' queries and document collection conversions, and continues to compute numeric scores on how

well each translated document representation will satisfy the given queries. After gathering all the scores, the model ranks the documents accordingly. In the end, users are able to read the original documents returned from the ranked document representations. In this thesis, I mainly focus on the examination of the fundamental IR weighting problem, and the proposal of new function models to promote the overall retrieval performance, more specifically, to find more relevant documents.

Ideally, when a user sends out a query, an IR system is supposed to be able to return a list of documents with a percentage accuracy in terms of its relevancy to the given query. In other words, the system should be equipped with the ability to estimate the exact relevancy of each document in the collection, and ideally should generate a “perfectly matched” list of documents to users. However, in practice, relevance in terms or documents is always difficult to calculate.

An effective term weighting mechanism, which has always been the most critical part of an information retrieval system, could help solve the problem. Almost all present retrieval models determine the degree of importance of a term to a document according to three variables [1]: (1) within document term frequency, (2) document length, and (3) the specificity of the term in the document collection. Typically, within document term frequency and document length will work together to tell the saliency of a term to a document. And the term specificity will be applied to reward documents with

rare terms in the collection, when a query has more than one term. It is worth noting that the concept of term frequency- inverse document frequency (tf-idf) is usually addressed and examined in IR papers as well.

Term frequency is a numerical statistic concept, applied in term weighting schemes of text mining, to calculate how important a term is to the selected document. A term frequency simply suggests that the weight of a term occurring in a document is simply proportional to the term frequency. Whereas the inverse document frequency implies that the specificity of a term can be quantified as an inverse function of the number of documents in which it occurs. Putting the *tf-idf* together, the value will increase proportionally to the number of times a term appears in the document, but the effect will be balanced out by the frequency of the term appearing in the document collections.

Generally, all different types of information retrieval models have grown over the past decades. Researchers have proposed several major models over years, and the current well-developed retrieval models, thus, could be categorized to four major groups based on the same propositional logic and different term weight estimation principles, the Boolean Model, the Vector Space Model, the Probabilistic Model, and the Language Model. All these models serve to calculate the relevance scores between queries and documents.

The Boolean model [30] is the first IR model based on Boolean logic, and the retrieving process mainly relies on whether the documents contain the given query terms. A vector space model requires both queries and documents to be converted in vectors of terms. It typically gives each query and document a dimensional vector, which is assigned with non-binary weights to the query and document index terms, and thus to compute the similarity between a query and documents in the collection [31]. The result generated is a list of documents ranked according to similarity to the user's query. Both of the two models introduced above are important in the history of the IR field since they opened up this grand new field of research.

The probabilistic model [32] then came in, and since then, it has taken the dominant role in IR system. The probabilistic model gives term weights based on the traditional probability theory. It ranks documents in decreasing order of probability of relevance to the information need, and classifies them to either relevant or non-relevant groups. Last but not least, a language model devotes to find out the probability of a document D generating an observed query Q [35]. That is to say it introduces the idea that if the document model is likely to generate the query [18], a document is a good match to the given query. In general, these four different existing models build up the majority of the family of the current information retrieval systems.

In the context of this thesis, I aim to propose three new influence functions to integrate to the probabilistic modelling, for the purpose of further enhancing the performance of the information retrieval system. Therefore, it is necessary to research beyond the background knowledge of the probabilistic model specifically.

In the past three to four decades, researchers and scholars have published extremely extensive and densely technical literature on the probabilistic approach to the IR system. Maron, Kuhns [36] and Miller [37] are the first researchers who attempted to develop the probabilistic theory in this field. They proposed the concept of “Relevance”, and applied indexing to collect, parse and store data easier, faster and more efficient. Since then, there has been a steady growth of the probabilistic modelling. All those well-known operational systems, such as Okapi, Indri, Lemur and Terrier, have been continuously improved by the dedicated work of researchers over the years.

Probabilistic modelling is any form of modelling which applies the presumed probability distribution of given input assumptions to generate the output result and its implied probability distribution. It provides a solution to compute the relevance certainty better, thus to return a list of relevant documents to satisfy the user’s query.

In the history of the development of probabilistic modelling, the indexing model is the first model introduced to index documents for the convenience

of searching by Maron and Kuhns [36]. They proposed the model with a key concept of “Relevance” to measure how relevant certain documents will satisfy the given query, and thus to compute the probability of each document in the collection.

The binary independence retrieval model is first influential model in IR system initially proposed by Robertson and Sparck Jones [40]. The basic assumption lying behind the model is that, following the cluster hypothesis, which closely associated documents tend to be relevant to the same requests, document terms should be distributed differently within relevant and non-relevant documents. Opposing to Robertson and Sparck Jones’ point of view, Cooper [49] in 1995 pointed out that the idea of binary independence is not the basis of this model, rather, it is a relatively weak assumption of linked dependence introduced by Fuhr [32] and Crestani [43fu].

Building on top of the binary independence retrieval model, the binary independence indexing model was proposed by Fuhr and Buckley [50] in 1991. The model has a significant advantage of that instead of specifying the document representations, the document representations are observed according to the given query terms. A binary vector has a value of either o or 1 to represent if a document contains a query term or does not. However, depending on the knowledge we have now, it is inappropriate to apply to the

information retrieval system in practice, since query terms are not purely independently existed in the context of documents.

Robertson et al. further studied the original and modified models above by integrating the probabilistic indexing model [36] into the binary indexing relevance model [40], and thus a unified model was proposed. Similar to all the baseline models used in the unification, the unified model assumes all document and query terms in the document collection exist independently, which we have known that with the application of cross term, better retrieving results could be generated [28].

Biebricher [60] and Fuhr [61] in the late 80s developed a description-oriented indexing approach named Darmstadt Indexing, to further divide the indexing task in description and decision. The approach modified the definition of relevance description, which makes the representation of documents more flexible and applicable in various circumstances.

Bookstein and Swanson's 2-Poisson Model [51] is proposed in purpose of improving document representation by deciding whether an index term should be assigned to a document if two document classes are produced.

1.2 Motivation

The probabilistic model of information retrieval has been one of the most popular models in the past fifty to sixty years around the world. It is not only comparatively good at solving traditional IR system problems of the

unclearness of users' information needs, and the ineffectiveness of returning conclusive lists of relevant documents; the probabilistic model also helps to overcome obstacles embedded in the nature of document relevancy, being inherently uncertain. However, there are still many problems in the field need to be further studied and explored.

First, almost all existing models in information retrieval system employ a single term frequency normalization mechanism. This mechanism does not take different aspects of a term's saliency within a document into account. For example, assume we have a term of "emergency in Toronto" in a given document. The frequency of "emergency in Toronto" in the document relative to the frequency of the other terms in the same document gives users very important implication of relevancy, which could not be achieved by the traditionally adopted document length based normalization scheme. Contrarily, the drawback of the relative frequency based term weighting could be covered by a length based normalization mechanism, which is able to restrict the retrieval system to return extremely long documents.

Second, another major limitation of the current models in the retrieval system is that they are incapable of balancing well between short and long documents. It will thus result in the system to retrieve a list of ranked relevant documents with low quality, when a mixture of short and long queries is presented. Depending on whether the static value of the parameter is set to be small or large, the models could perform better either

in short or long documents. When users enter shorter queries, if a weighting scheme prefers short documents, it will pull up extremely short documents since these are the documents with lower verbosity level and matches the queries better. Whereas when shorter queries encounter long document preference, the overall retrieval performance may degrade.

Third, most of the present models usually employ their own single functions, and apply the proposed single function to the document collection. For the purpose of achieving better results, the application of multiple sources of functions in the retrieving process may avoid single function limitations, while exploit the advantages of each function to the fullest extent.

1.3 Main Contributions

In this thesis, new perspectives of approaching to the problems above will be introduced. This thesis serves to enhance the performance of the traditional probabilistic models, while to deploy the probabilistic ranking principle into more domains. In particular, I replace the term frequency to relative term frequency in the classic BM25 function, and focus on proposing three new kernel functions to integrate relative term frequency into the traditional probabilistic function, BM 25 to address the fundamental IR weighting problems, and to minimize the existed weaknesses. All the

experimental results have shown that the proposed new methods have achieved an overall better performance.

In general, the new methods could be categorized into two main approaches. In the relative term frequency approach, the new model will apply the term frequency normalization scheme considering relative single term frequency weighting and the term frequency normalization based on document length, instead of the employment of a traditional single term frequency normalization mechanism. There are two major factors affecting the term frequency normalization, namely the relative intra-document term frequency and the length regularized term frequency.

For the relative intra-document term frequency, the significance of a term is measured by considering its frequency relative to the average term frequency of the document [1]. According to the formula proposed by Paik [1],

$$Rel\ tf = \frac{tf(t,D)}{avgtf(D)}, \tag{1}$$

where $tf(t,D)$ denotes the frequency of term t in the document D and $avgtf(D)$ represents the average term frequency of t in D, would be too much in favor of long documents, since the denominator will be very much close to 1 as it meets a long document. On the other hand, the length

regularized term frequency adopts the term frequency normalization by considering the number of certain terms in a document. With the assumption of average document length of the collection and unchanged frequency of certain terms appearing in the average length document, the length dependent normalization would be achieved with an equation of

$$LRTF\ (t,\ D) = TF\ (t,\ D) \times log_2\left(1 + \frac{ADL\ (C)}{len\ (D)}\right) \qquad (2)$$

favoring short documents. Thus, one component of the term frequency tends to favor short documents, while the other component favors long documents. This structure of relative term frequency normalization keeps a good balance in preferences, and thus is able to calculate the frequency of a certain term in document with more accuracy.

The highlight of this thesis is embedded in the second approach. With the application of the new solution of capturing and modelling the influences of relative term frequencies among different terms in a document, the study will propose three novel kernel functions to help the traditional probabilistic BM25 model to achieve better retrieval results. In other words, this thesis will try two methods of enhancement. First, it will try to replace the term frequency with the relative term frequency concept in the original BM25 weighting function; besides, it will also integrate the three newly defined kernel functions, a linear, a quadratic and a cube served for relative term

frequency calculation, into the well-known classical BM25 model to form linear combines. In this way, a combination of the two sets of functions will be able to effectively enhance the performance of probabilistic information retrieval model.

1.4 Thesis Outline

This thesis will consist of seven chapters, appendices and a list of references. The organization of the thesis is as follows. In Chapter 1, I focus on presenting an overview of the information retrieval system, including the major IR models, and pointing out the motivation and contribution behind this thesis. Besides, a structure of this thesis is given. In Chapter 2, a group of prior work related to the study are discussed. The topics mainly include the present fundamental information retrieval (IR) weighting models, and the “relevance” versus “relevance in probabilistic modelling”. Chapter 3 introduces the new approaches and the experimental methodologies adopted in this thesis specifically. And these three newly proposed kernel functions are used to examine the effect of relative term frequency when integrating into the BM25 probabilistic function. More specifically, in Section 3.1, the three newly proposed influence functions will be presented; and the Section 3.2 discusses how the relative term frequency replaced the term frequency in BM25 and how the influence functions will be integrated into BM25, and thus to enhance the probabilistic information retrieval model by rewarding

query terms with high relative term frequency. In Chapter 4, the major components of the information retrieval environment used in the thesis are described. These mainly include the system, the data collections, gold standard and the evaluation metrics used in the experiments. The Chapter 5 is an empirical study of the experimental results, while the Chapter 6 presents the analysis and discussion of the experiments. Chapter 7 concludes the thesis and gives possible directions for future work.

Chapter 2 Literature review

The study of information retrieval was originated from the library science field in early 1960s according to Maron and Kuhns's paper [36]. With the help of indexing each document with a set of weighted keywords, an information retrieval system serves to retrieve all "relevant" documents by comparing users' query representations and document representations.

The literature review presents other research work related to this thesis. In this chapter, the thesis will describe the present fundamental term weighting models in Information Retrieval first. In Section 2.1, the very first Boolean model is introduced, following by the vector space model to be presented in 2.2. Then the Section 2.3 will focus on reviewing the related works on the probabilistic models, including the probability ranking principles and BM25. Section 2.4 states the language modelling approach in the information retrieval system. Last but not least, the relevance and relevance in probabilistic modelling is discussed and reviewed in Section 2.5.

2.1 The Boolean Model

The Boolean model and the Vector Space model are the two earliest well-known weighting models in history of information retrieval. Although they have certain disadvantages in weight assignment, many present popular models were extended on top of these two “ancestors”.

The Boolean model [30, 63] is built based on Boolean logic and classic set theory. The model will retrieve a document if and only if the information in the document is an exact match to the user’s query. Query terms of the Boolean model are connected with three basic logical operators, the logical product of AND, the logical sum of OR and the logical difference of NOT. However, since the unconstrained NOT notion is very expensive in the retrieving process, in most cases, the system will not include it as one the operators.

The advantages of the Boolean are that the system is very efficient, predictable, and it works very well when users know exactly what they need to extract. However, most people find it difficult to create a good query for the retrieval process. The precision and recall usually have strong inverse correlation. Besides, all terms are weighted as equally important. Moreover, documents under the Boolean model are assigned the weight of either 0 or 1, meaning that those documents which are close to the given queries will be all rejected. As a result, the list of documents retrieved will either be too few or too many.

2.2 The Vector Space Model

Similar to the Boolean model, the vector space model [44, 64] has a long history in the information retrieval field as well. The model is fundamental to a host of information retrieval operations, ranging from the score calculation of documents on a given query, document classification and document clustering.

The basic matching theory of the model is that the value of a document in the vector will be non-zero, as long as there is at least one term appears in it. The vector space model procedure could be grouped into three stages. The first stage is to index the document where the contents containing terms are extracted from the document text. Both a set of documents and users' queries are represented in the form of vectors in a common vector space as shown below.

$$Q = (w_{1,Q}, \ldots\ldots, w_{n_{v,}Q}) \quad (3)$$

$$D = (w_{1,D}, \ldots\ldots, w_{n_{v,}D}) \quad (4)$$

Where *w* is the weight for a dimension and n_v is the dimensionality of the vectors.

The second stage is to assign non-binary weights to the indexed terms based on the formulae (3), (4), (5) and (6) to enhance the generated results of the retrieved relevant documents. The third stage then ranks the list of documents according to the query similarity measure, where the similarity of a document vector to a query vector is the cosine of the angle between them, in equation (7). The similarity functions include but are not limited to,

The Inner Product:

$$Q \cdot D = \sum_{i=1}^{n_v} w_{i,Q} \times w_{i,D} \quad (5)$$

The Cosine Similarity:

$$Cosine\ (Q,D) = \frac{\sum_{i=1}^{n_v} w_{i,Q} \times w_{i,D}}{\sqrt{\sum_{i=1}^{n_v} w_{i,Q}^2} \times \sqrt{\sum_{i=1}^{n_v} w_{i,D}^2}} \quad (6)$$

The Dice Similarity:

$$Dice\ (Q,D) = \frac{2\sum_{i=1}^{n_v} w_{i,Q} \times w_{i,D}}{\sum_{i=1}^{n_v} w_{i,Q}^2 + \sum_{i=1}^{n_v} w_{i,D}^2} \quad (7)$$

The Jaccard Similarity:

$$Jaccard\ (Q,D) = \frac{\sum_{i=1}^{n_v} w_{i,Q} \times w_{i,D}}{\sum_{i=1}^{n_v} w_{i,Q}^2 + \sum_{i=1}^{n_v} w_{i,D}^2 - \sum_{i=1}^{n_v} w_{i,Q} \times w_{i,D}} \quad (8)$$

$$Sim\ (d_i, q) = cos\theta \tag{9}$$

The major advantages of the vector space model are that the model is simple, effective, and it is able to incorporate any kind of term weights. Besides, it computes a continuous degree of similarity between queries and documents. In other words, instead of being either an exact match or not match at all to the given query, the model allows partial matching that the term weights are not valued binary [28]. However, the biggest drawback of the model underlies that terms are still assumed to be in the independence relationship. Moreover, the values to the vector components are not appropriately defined [30], and the model lacks justification for some vector operations as well.

It is worth noting that an important concept, a term frequency-inverse term document frequency (tf-idf) [69] was introduced by Salton et al. in the classical vector space model in 1983. Term frequency (tf) describes how well a term describes its document by calculating the frequency of its appearance in a document; while the inverse document frequency (idf) refers to terms that occur in many documents of a collection are less useful for discriminating among documents. The document frequency, or the number of documents containing the term idf, is often calculated as,

$$I = \log\frac{N}{df} + 1 \quad (10)$$

Combining these two factors together, the

$$w_{d,t} = tf_{d,t} \times idf_t \quad (11)$$

weighting scheme becomes the most common term weighting approach for the vector space model.

2.3 The Probabilistic Model

The probabilistic model in IR was developed following the basic probabilistic theory. The probability of a document D being relevant to the user's query Q is represented by

$$P(R|Q,D) = \frac{P(D|R,Q)*P(R|Q)}{P(D|Q)} \quad (12)$$

In general, probabilistic models avoid shifting uncertainties to users by providing solutions to compute relevance certainty, whereas the Boolean model searches the documents based on Boolean logic and classical set theory that uncertainties remain as uncertainties for users to deal with [45]. While in comparison to the vector space model whose documents are

retrieved and ranked by the degree of similarity, probabilistic models are more interpretable and computable.

2.3.1 Probability Ranking Principle (PRP)

The probability ranking principle was introduced by Robertson in the late 1970s [66]. The principle has been served as the basis of most probabilistic approaches in IR, which states specifically that "if a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, the overall effectiveness of the system to its users will be the best that is obtainable on the basis of those data." Mathematically, this PRP is defined as,

$$C \cdot P\left(R\middle|q_k, d_j\right) + \bar{C} \cdot \left(1 - P\left(R\middle|q_k, d_j\right)\right) \leq C \cdot P(R|q_k, d_m) + \bar{C} \cdot \left(1 - P(R|q_k, d_m)\right) \quad (13)$$

Where *C* is the cost of retrieving a relevant document and $\bar{C}$ is the cost of retrieving an irrelevant document. d_j and d_m denote the two different document candidates.

2.3.2 The Binary Independence Retrieval Model

The binary independence retrieval model [40] is the model that has traditionally been used with PRP, and it develops the idea with precise assumptions shared by most of other probabilistic models. The “binary” here introduces the idea equivalent to Boolean, where the documents and queries are both represented in binary term incidence vectors. And the “independence” here suggests that the terms occurring in the documents share no association between each other, and thus are modelled independently.

More specifically, we model the probability that a document is relevant to the query with the term incidence vector $P(R|\vec{x},\vec{q})$. With the application of Bayes theorem, the probability of relevance is:

$$P(R=1|\vec{x},\ \vec{q})=\ \frac{P(\vec{x}|R=1,\vec{q})*P(R=1|\ \vec{q})}{P(\vec{x}|\vec{q})} \tag{14}$$

$$P(R=0|\ \vec{x},\vec{q})=\ \frac{P(\vec{x}|R=0,\vec{q})*P(R=0|\ \vec{q})}{P(\vec{x}|\vec{q})} \tag{15}$$

Here we denote $P(\vec{x}|R=1,\vec{q})$ as the probability of a relevant document being retrieved while denote $P(\vec{x}|R=0,\vec{q})$ as an irrelevant, besides, the $\vec{x}$ will

be the document representation. According to the classic probabilistic theory, we must have,

$$P(R = 1|\,\vec{x},\vec{q}) + P(R = 0|\,\vec{x},\vec{q}) = 1 \qquad (16)$$

In order to derive a ranking function for query terms, we set O as a constant for the given query, and thus,

$$O(R|\,\vec{x},\vec{q}) = \frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})} = \frac{\frac{P(R=1|\vec{q})*P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|\vec{q})}}{\frac{P(R=0|\vec{q})*P(\vec{x}|R=0,\vec{q})}{P(\vec{x}|\vec{q})}} = \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \times \frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})} \qquad (17)$$

In Cooper's paper in 1995 [49], he suggests that the binary independence assumption made by Robertson and Sparck Jones [40] in 1976 was inappropriate. He believes the assumption behind the binary independence retrieval model should be the linked dependence of the form,

$$\frac{P(\vec{x}|R=1)}{P(\vec{x}|R=0)} = \prod_{i=1}^{n} \frac{P(\vec{x}|R=1\,)}{P(\vec{x}|R=0\,)} \qquad (18)$$

After transformation and simplification,

$$\frac{P(R=1|\,\vec{x},\vec{q})}{P(R=0|\,\vec{x},\vec{q})} = \frac{P(R=1|\,\vec{q})}{P(R=0|\,\vec{q})} \times \frac{P(\vec{x}\,|R=1,\vec{q}\,)}{P(\vec{x}\,|R=0,\vec{q}\,)}$$

$$= \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \times \prod_{i=1}^{n} \frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q}\,)}$$

$$= \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \times \prod_{i=1}^{n} \frac{P(\vec{x}=1|R=1,\vec{q})}{P(\vec{x}=1|R=0,\vec{q}\,)} \times \prod_{i=1}^{n} \frac{P(\vec{x}=0|R=1,\vec{q})}{P(\vec{x}=0|R=0,\vec{q}\,)}$$

(19)

To further the simplification of the equation $\frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})}$, we denote R=1 to *R*, R=0 to $\overline{R}$, $p_i = P(\vec{x}\,|R,\vec{q})$ and $q_i = P(\vec{x}\,|\overline{R},\vec{q})$, meanwhile, T = $\{t_1,\ldots,t_n\}$ is set as the set of terms in the collection, and d_i denotes the document retrieved judged to be relevant to the given query.

For the terms that do not occur in the set of $\vec{q}$, $p_i = q_i$. Thus, the equation above is transformed as,

$$\frac{P(R=1|\,\vec{x},\vec{q})}{P(R=0|\,\vec{x},\vec{q})} = \frac{P(R=1|\,\vec{q})}{P(R=0|\,\vec{q})} \times \prod_{t_i \in d_i \cap \vec{q}} \frac{p_i}{q_i} \times \prod_{t_i \in \vec{q} d_i} \frac{1-p_i}{1-q_i}$$

$$= \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \times \prod_{t_i \in d_i \cap \vec{q}} \frac{p_i\,(1-q_i)}{q_i(1-p_i)} \times \prod_{t_i \in \vec{q} d_i} \frac{1-p_i}{1-q_i} \quad (20)$$

Based on the relevance feedback solution, we make f to be the user selected number of documents, f_i will thus be the number of documents among f where term t_i occurs; and r to be the documents judged as relevant to the given query, r_i will thus be the number of relevant documents retrieved where term t_i occurs. And finally, the probabilities could thus be calculated by,

$$p_i \approx \frac{r_i}{r} \tag{21}$$

$$q_i \approx \frac{f_i - r_i}{f - r} \tag{22}$$

2.3.3 The Binary Independence Indexing Model

Unlike the binary independence retrieval model which focuses on the relation between a single query and the whole document collection, the binary independence indexing model looks at a document in relation to a number of queries. The idea of this model originated from the indexing model proposed by Maron and Kuhn in the 1960s [36].

The model seeks to find the probability of relevancy between the set of query and a document by assuming a binary vector z, whose value is equal to 1 when a term occurs in the query, otherwise equals to 0. With the

application of Bayes theorem, the binary independence indexing model could thus to be written as,

$$P(R|z,d') \tag{23}$$

$$P(R|z,x) = \frac{P(R|x)* P(z|R,x)}{P(z|x)} \tag{24}$$

Where d' denotes the document categorized to be relevant to the query representation z, x as the notion for document, $P(R|x)$ is the probability of a document $x$evaluated to be relevant to the query, and $P(z|R,x)$ is the probability of the document being relevant to the query representation z. Since the query representation and the document representation are assumed to be independent from each other, the $P(R|z,x)$ could be further simplified as,

$$P(z|R,x) = \prod_{i=1}^{n} P\,(z_i|R,x) \tag{25}$$

2.3.4 2-Poisson Model and N-Poisson Model

In 1974, Bookstein and Swanson [51] first proposed a 2-Poisson indexing model for term frequencies to suggest some ideal ways of incorporating certain variables into the probabilistic models. Robertson and Walker in 1994, further studied the topic and proposed that within document

term frequency, document length and the within query term frequency are the three major variables concerned for the weighting function [14],

$$w\left(\underline{x}\right) = \log\frac{P\left(\underline{x}|R\right)P\left(\underline{0}|\overline{R}\right)}{P\left(\underline{x}|\overline{R}\right)P\left(\underline{0}|R\right)} \quad (26)$$

Where $\underline{x}$ is the vector of information about the document, R denotes the relevance, $\overline{R}$ is the non-relevance, and $\underline{0}$ is the reference vector representing the zero-weighted document.

The 2-Poisson model is constructed based on the assumption that by determining which one of the two Poisson distributions the selected term should belong to (choose one in between), it will be possible to decide whether a term should be assigned to a document. The two Poisson distributions which mentioned above are modelled in the distribution of within document frequencies for relevant documents, and the distribution of within document frequencies for non-relevant documents with a different mean. The problem with this two-Poisson model is that it needs probabilities conditioned on relevance. According to Robertson and Sparck [40], their final formula has too many parameters which no data could practically fit into them.

$$P\left(TF_i = tf|rel\right) = p_{i1}E_{i1}\left(tf\right) + (1 - p_{i1})E_{i0}(tf) \quad (27)$$

Where E_{i1} denotes the eliteness in Robertson's paper, referring to a relevant term, and the E_{i0} referring to the irrelevance of certain term. Then the probability function (27) leads to an equation for the term weighting of an elite term i:

$$w_i^{elite} = \log \frac{(p_1 E_1(tf)+(1-p_1)E_0(tf))(p_0 E_1(0)+(1-p_0)E_0(0))}{(p_1 E_1(0)+(1-p_1)E_0(0))(p_0 E_1(tf)+(1-p_0)E_0(tf))} \quad (28)$$

Where E and $\overline{E}$refers to the eliteness and non-eliteness respectively. Eliteness could be interpreted as a form of aboutness: if the term is elite in the document, the document is considered to be about the concept.

The n-Poisson model is an extension of the 2-Poisson model to the n-dimensional case, assuming that there are n classes of documents in which the term t appears with different frequencies.

2.3.5 The BM25

The Best Match 25 (BM25) model is a non-binary model originated as part of the Okapi Basic Search System in the TREC Conferences. As discussed above, the three major principles, which are the inverse document frequency, term frequency, and document length normalization, compose an overall very well performed term weighting scheme.

However, classic probabilistic models only cover the inverse document frequency principle. Therefore, Stephen Robertson and some other researchers tried to propose a BM25 on top of their previously proposed BM1, BM11 and BM15 models, to form a direct extension of the classic probabilistic model which serves to cover all three major principles listed above.

Initially, the Okapi system used the BM1 formula in probabilistic model as a ranking formula when non-relevance information is given,

$$sim\left(d_j, q\right) \sim \sum_{k_i \in q \hat{} k_i \in d_j} \log \frac{N - n_i + 0.5}{n_i + 0.5} \tag{29}$$

Then, based on the 2-Poisson model of term occurrence within the documents, the concept of term frequency is integrated into the BM1 ranking formula, by

$$S_1 \times \frac{f_{i,j}}{K_1 + f_{i,j}} \tag{30}$$

Where S_1 is a scaling constant, $f_{i,j}$ is the frequency of the occurrence of term k_i within document d_j, and $K_1$is another constant setup for each collection in the experiments. When $K_1$is 0, this whole factor will value 1, and thus bears no effect in the ranking.

By adding the document length normalization principle into the formula above, this equation is transformed to,

$$S_1 \times \frac{f_{i,j}}{\frac{K_1 \times len\,(d_j)}{avg\_doclen} + f_{i,j}} \tag{31}$$

Where $len\,(d_j)$ is the document length of d_j, and avg_doclen denotes the average document length of the document collection.

Aside from that, the BM1 function also used a correction factor which depends purely on the document length and query length,

$$K_2 \times len(q) \times \frac{avg_{doclen} - len\,(d_j)}{avg_{doclen} + len\,(d_j)} \tag{32}$$

Where K_2 is another constant applied, and the $len(q)$ is the length of the query.

Thus, the last step is to apply the third factor to the term frequencies within queries,

$$S_3 \times \frac{f_{i,q}}{K_3 + f_{i,q}} \tag{33}$$

Where S_3 again, is the scaling constant related to K_3, which is a constant as well. And the $f_{i,q}$ is the frequency of term k_i within query q.

With the integration of the three factors introduced above, the BM1 function was led to BM 11, and BM15 respectively, where $k_i[q, d_j]$ is a short notation for $k_i \in q \hat{} k_i \in d_j$.

$BM11$:

$$sim\left(d_j, q\right) \sim G_2 + \sum_{k_i \in q \hat{} k_i \in d_j} \frac{S_1 f_{i,j}}{\frac{K_1 \times len\left(d_j\right)}{avg\_doclen} + f_{i,j}} \times \frac{S_3 f_{i,q}}{(K_3 + f_{i,q})} \times \log \frac{N - n_i + 0.5}{n_i + 0.5} \quad (34)$$

$BM15$:

$$sim\left(d_j, q\right) \sim G_2 + \sum_{k_i \in q \hat{} k_i \in d_j} \frac{S_1 f_{i,j}}{(K_1 + f_{i,j})} \times \frac{S_3 f_{i,q}}{(K_3 + f_{i,q})} \times \log \frac{N - n_i + 0.5}{n_i + 0.5} \quad (35)$$

The widely used BM25 ranking formula we use today is structured by combining the BM11 and BM15 ranking formulae, mainly the term frequency factors:

$$S_1 \times \frac{f_{i,j}}{K_1((1-b) + b\frac{len\left(d_j\right)}{avg_{doclen}}) + f_{i,j}} \quad (36)$$

The function above provides a combination of the B11 and B15, where *b* is a constant that values between 0 and 1. When *b* is equal to 1, the equation is reduced to the B11 term frequency factors, while when *b* is equal to 0, it is reduced to the B15 term frequency. And thus, the BM25 is presented as follows,

$$W = \frac{(k_1+1)*tf}{K+tf} * \log\frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} * \frac{(k_3+1)*qtf}{k_3+qtf} \oplus k_2 * nq * \frac{(avdl-dl)}{(avdl+dl)} \quad (37)$$

where *w* is the weight of a query term, *N* is the number of indexed documents in the collection, *n* is the number of documents containing a specific term, *R* is the number of documents known to be relevant to a specific topic, *r* is the number of relevant documents containing the term, *tf* is within-document term frequency, *qtf* is within-query term frequency, *dl* is the length of the document, *avdl* is the average document length, *nq* is the number of query terms, the $k_i s$ are the tuning constants which are empirically determined according to the database and on the nature of the queries, K equals to $k_1 * ((1 - b) + b * dl/avdl)$, and $\oplus$ indicates that instead of applying to each term, the component following will be added only once per document.

The BM25 has been one of the most efficient and widely-used information retrieval weighting models in the past three decades. Unlike other probabilistic models, the BM25 could be computed without relevance information. Since BM25 almost outperforms classic vector model for general data collections, it has substituted the vector space model to be used as a baseline for comparison for decades.

2.4 The Language Modelling Approach

Ponte and Croft [35] proposed a method to improve the quality of search results with the application of the indexing models. The basic working function is that users will be able to inquire about the probability of a given user query to be generated by a given indexing model for a set of documents. In other words, the probability distributions of terms could be used to determine the probability of observing a given user query. The language model, which is the probability distribution discussed above, presents a variant of the idea of using the distribution of index terms in the collection as the basis for ranking. And the statistic-based language model has been one of the dominant IR weighting models as well.

Given a document d_j with M_j being a reference to the language model for that document, a simple estimate of term probabilities will be,

$$P(k_i|M_j) = \frac{f_{i,j}}{\sum_i f_{i,j}} \quad (38)$$

Where we set the $P(q|M_j)$ as the probability of generating a user's query from the language model of document d_j. With the assumption of index terms independence, we could compute $P(q|M_j)$ from $P(k_i|M_j)$. To solve the problem of not allowing partial matches, the formula above could be rewrite as,

$$P(k_i|M_j) = \begin{cases} \frac{f_{i,j}}{\sum_i f_{i,j}} & if\ the\ value\ of\ f_{i,j}\ is\ greater\ than\ 0; \\ \frac{F_i}{\sum_i F_i} & if\ the\ value\ of f_{i,j}\ is\ equal\ to\ 0. \end{cases} \quad (39)$$

2.5 Relevance and Probabilistic Relevance

Hjørland and Christensen [62] set the definition of "relevance" in the context of information retrieval and library science. They suggest that "something (A) is relevant to a task (T) if it increases the likelihood of accomplishing the goal (G), which is implied by T" [62]. Indeed, Relevance (R) is used to describe the relationship between users' queries and document collections. If the users are able to find the information they

requested by the queries in the data collection, we say the documents are relevant to the given queries.

In probabilistic modelling, the relevance is usually calculated from a single or a set of computable functions with multiple variables based on the users' specific requests for certain information. When user assigns a query to retrieve certain information, the retrieval system tries to rank documents according to the degree of relevance. Most of the systems will order the documents based on the numeric scores assigned specifically to the documents.

Some researchers in this field have done some studies to improve the performance of the information retrieval models using the concept of relative term frequency. Singhal [16] measures the importance of a term in a document by considering its frequency relative to the average term frequency of the document. The formula of the relative term frequency of a term, *t*, is,

$$Rel\ tf = \frac{tf(t,D)}{avgtf(D)} \tag{40}$$

where *tf(t, D)* is the frequency of term *t* in document *D*, and *avgtf(D)* denotes the average term frequency of all terms in *D*. We could tell from equation (40) that it has a strong preference to excessively long documents, since the denominator *avgtf(D)* will get close to 1 when the document is long.

In order to effectively mitigate the preference effect, a sub-linear transformation of equation (40) is suggested to penalize the excessively long documents, and to normalize the value of term frequencies according to the number of unique terms in the document, as shown in the equation (41) below.

$$Rel\ tf = \frac{log2(1 + tf(t,D))}{log2(1 + avgtf(D))} \quad (41)$$

Similarly, another key factor influencing the normalization of the term frequency is the length regularized term frequency[1]. It takes the number of unique terms present in the document into consideration. Furthermore, the factor assumes that documents in collection should be in average document length and the frequency of the terms appearing in the average length document should keep steadily unchanged.

According to the baseline function of the length regularized term frequency [1],

$$TF\ (t,\ D)\ \times \frac{ADL\ (C)}{len\ (D)} \quad (42)$$

where *ADL(C)* is the average document length of the collection and *len(D)* is the length of the document *D*, the excessively long documents will be over-

penalized, as the increase in term frequency would not follow a linear relationship with the document length. In order to balance out this biased preference,

$$LRTF\ (t,\ D) = TF\ (t,\ D) \times log_2\left(1 + \frac{ADL\ (C)}{len\ (D)}\right) \qquad (43)$$

was employed by Paik [1] to solve the problem.

For the purpose of qualifying the saliency of the query terms in TF-IDF model[2], Paik [1] presents a novel TF-IDF term weighting scheme with equation (42) and (43), which employ two different within document term frequency normalizations to determine the importance of a term in the context of a certain document.

Ye [17] further studied the topic with an application to the Pseudo Relevance Feedback. In his paper, he finds out that, traditionally, most of the existing models employ the single term frequency normalization criteria and mechanism that do not take the various aspects of a term's saliency in the feedback documents into account. To address the issue, he proposed a relatively simple and effective model with the relative term frequency transformation method based on equation (42). The model thus helps to capture the saliency of a candidate term associated with the original query terms in the scenario of Pseudo Relevance Feedback.

Lv [10] notes in his article that it is a common deficiency in current retrieval models that the document length, a component of term frequency normalization, is not lower-bounded properly. Moreover, the extremely long documents tend to be overly penalized. In order to provide feasible solutions to address these problems, Lv and his co-authors introduce a sufficiently large lower bound for term frequency normalization. In another paper written just a year later[8], Lv suggests that the parameter k_1 in classical BM25 model, which is generally set to a term-independent constant, should be set in a term-specific way.

Chapter 3 Our Approach

In this chapter, I will propose a new approach which suggests that there should be a proportional increase of the effectiveness of the ranking system, as the relative term frequency being weighted at a higher score. In Section 3.1, I will propose three kernel functions to help to compute the relative term frequencies, and thus to see their impact. In the next Section 3.2, I will be doing two experiments, in which one would replace the term frequency in the traditional probabilistic term weighting function BM25 with the relative term frequency directly; and the other would integrate the relative term frequency into the BM25 function with a linear combination.

3.1 The Influence Functions

The intuition behind the approach of this thesis is that a query term with a high relative term frequency in a document will be more likely to reveal the meaning of the document, and thus the document will achieve higher rank in the collection. Hence, the newly proposed influence functions should reward the query terms with high relative term frequencies while penalize those with low relative term frequencies.

In this thesis, I define an influence function *IF(rtf), in which* the relative term frequency (*rtf)* should hold the following properties:

1. The relative term frequency should always be non-negative:
   That is $IF(rtf) \geq 0$, which means that the impact of a relative term frequency should always be a positive value;

2. The values of the relative term frequency should be continuous:
   The absolute value of$|IF(rtf) - IF(rtf + 1)|$ should always be small, which implies that there is only a slight difference existing between the two neighboring relative term frequencies;

3. The shape of the influence functions should keep monotonic:
   That is $IF(rtf) < IF(rtf + 1)$, which suggests that the influence function should always increase with the increase in the value of relative term frequency;

4. The influence functions should have clear identities:

   That is *IF(avgrtf) = 0*, which sets a clear rule that the influence function sets 0 to be the standard influence.

In order to examine the influence of the relative term frequency, we build three novel influential functions on top of the relative frequency properties introduced above. The three functions are constructed as follows which satisfy all the four properties listed above with different gradients,

$$IF_{Linear}(tf(t,D),\beta) = \beta * (\frac{tf(t,D) - avgtf(D)}{a * avgtf(D)}) \quad (44)$$

$$IF_{Quadratic}(tf(t,D),\beta) = \beta * (\frac{tf(t,D) - avgtf(D)}{a * avgtf(D)})^2 \quad (45)$$

$$IF_{Cube}(tf(t,D),\beta) = \beta * (\frac{tf(t,D) - avgtf(D)}{a * avgtf(D)})^3 \quad (46)$$

*when avgtf ≤ tf (t, d) ≤ (α+1) * avgtf,*

where *tf (t, D)* is denoted as the term frequency of query term *t* in document *D*; *avgtf(D)* represents the average term frequency of all terms in document D; α is a constant which is set to the value of 10 based on the experiment statistics in this thesis, and β is another parameter controlling the influence

of relative term frequency, whose range starts from 0 to 20. It is worth noting that when $\beta = 0$, the proposed influential functions (*IFs*) will not contribute to the ranking.

The following two properties described by equation (47) and equation (48) are also held for the proposed influence functions, meanwhile they are defined by the equation (44) to equation (46) presented above.

$$IF_{Linear}(tf(t,D),\beta) = \quad IF_{Quadratic}(tf(t,D),\beta) = \quad IF_{Cube}(tf(t,D),\beta) = \beta$$

$$\text{when } tf(t,d) > (a+1) * avg.TF \tag{47}$$

$$IF_{Linear}(tf(t,D),\beta) = \quad IF_{Quadratic}(tf(t,D),\beta) = \quad IF_{Cube}(tf(t,D),\beta) = 0$$

$$\text{when } tf(t,d) < avg.TF \tag{48}$$

In the *figure 2 below*, the shapes of these three influential functions, Linear, Quadratic and Cube, are given.

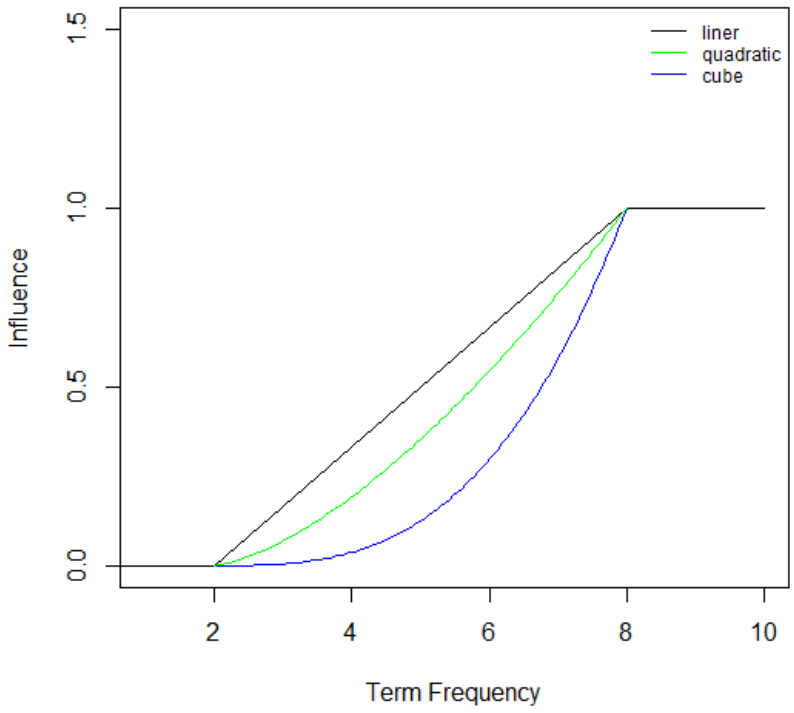

Figure 2. The shapes of the three influence functions on relative term frequency.

3.2 Integration into BM 25

Building on top of the listed three new influence functions proposed, this thesis will then do two tests: one will directly replace the term frequency with the relative term frequency concept in the classic BM25 function; and the other will try to integrate the influence functions into classic BM25 probabilistic model aiming to enhance the probabilistic information retrieval process by rewarding query terms with high relative term frequency.

In Robertson and Taylor's paper in 2004 [13], the authors studied how to integrate the field information into BM25. Similar to the idea of their approach, in the research of this thesis, I linearly combine the influence function of a certain term based on its relative term frequency. Hence, I use the term's within-document term frequency to create a comprehensive item of tf_{RTF}. It will be effective in characterizing its difference to other terms, and the representative function is the following presented function, shown as equation (49):

$$tf_{RTF}(t,\ D)\ =\ tf(t,D)\ +\ IF(tf(t,D),\ \beta) \tag{49}$$

where *IF* is one of the three influence functions defined above.

The classic BM 25 formula is presented below,

$$W\ =\ \frac{(k_1+1)*tf}{K+tf}\ *\ \log\frac{(r+0.5)/\ (R-r+0.5)}{(n-r+0.5)/\ (N-n-R+r+0.5)}\ *\ \frac{(k_3+1)*qtf}{k_3+qtf} \oplus k_2 * nq * \frac{(avdl-dl)}{(avdl+dl)} \tag{37}$$

With the simplest form of integration, replacing term frequency with $tf_{RTF}(t,\ D)$, the BM 25 is enhanced with this new concept as follows to become a new model, named BM25-RTF:

$$\text{BM25} - \text{RTF} = \frac{(k_1+1) * tf_{RTF}}{K + tf_{RTF}} * \frac{(k_3+1) * qtf}{k_3 + qtf} * \log \frac{(N-n+0.5)}{(n+0.5)} \qquad (50)$$

where *N* is the number of documents in the collection; *n* is the number of documents which contain q_i ; tf_{RTF} is the within-document term frequency combined with relative term frequency; *qtf* is the within-query term frequency; k_3 is a tuning constant defaulted to the value of 8; K is set to equal to k_1 * *((1-b) +b * dl / avdl); dl* is the length of the document; and lastly the *avdl* represents the average document length.

Another way of approaching to the probabilistic weighting function with relative term frequency is to try to integrate the influence functions into classic BM25 probabilistic model. For the purpose of enhancing the probabilistic information retrieval process, the functions will reward query terms with high relative term frequency, and will linearly integrate into the BM25.

Chapter 4
Information Retrieval Environment

In this chapter, I will introduce more details about my settings used to test the newly proposed methods for my experiments. The structure and workflow of my applied information retrieval system will be presented in Section 4.1 first. In Section 4.2, I will summarize the characteristics of the data collections I used for the experiments. In the following Section 4.3, the gold standard for the experimental results is given. And the last Section 4.4 will describe the evaluation metrics in details.

4.1 The Experimental Platform

In the experiments, I use the Okapi BSS (Basic Search System) [41, 53] as my major search system, and conduct the information retrieval experiments using the improved Okapi system [54, 55, 56, 57, 58, 53].

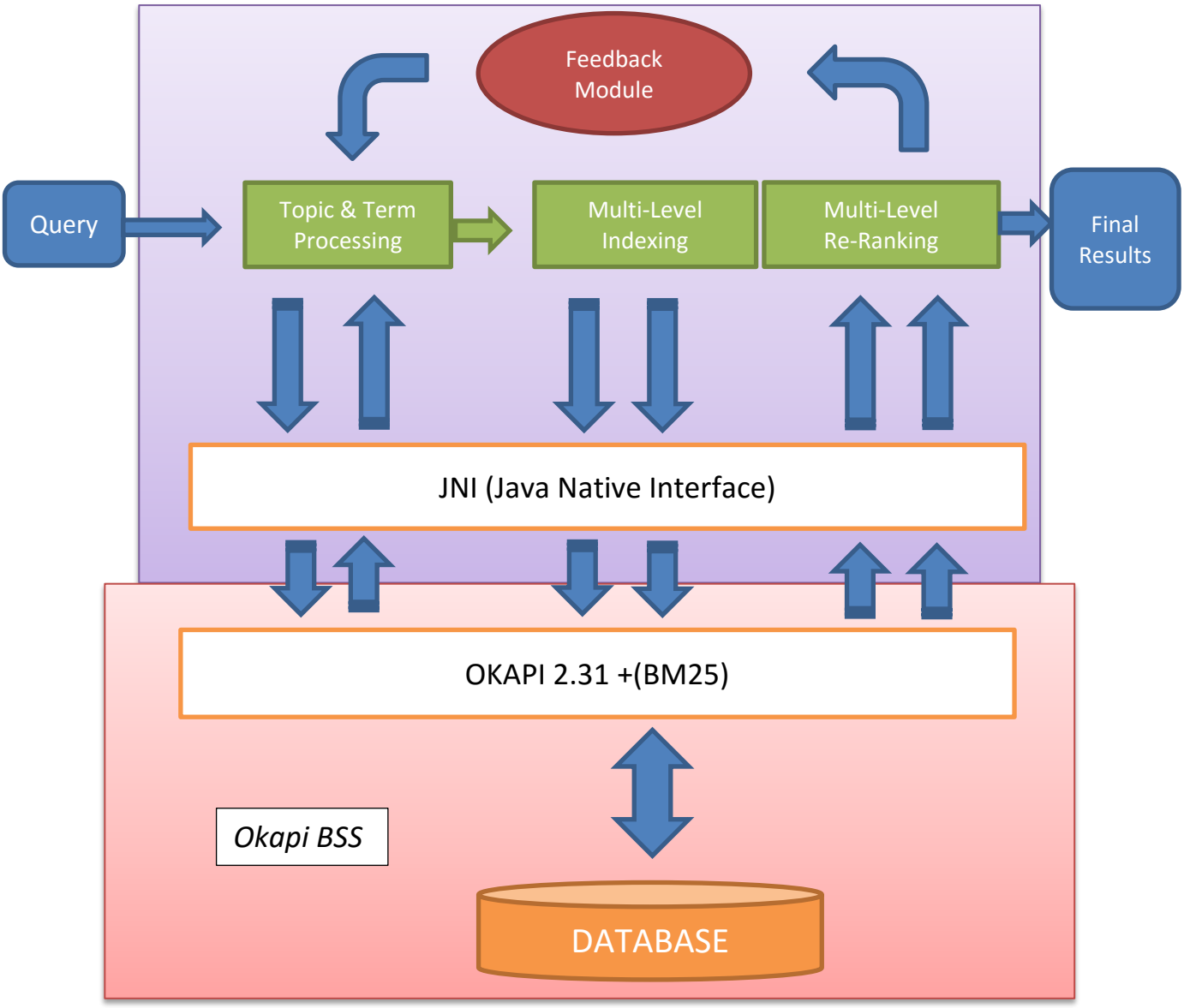

Figure 4. Experiment System Architecture

As shown in the Figure 4 above, the Okapi BSS is an information retrieval system build on the basis of the probability model introduced by Robertson and Sparck Jones [38, 14]. In information retrieval research and

experimentation field, it is one of the most well-established and well-performing systems that has been widely used. The retrieval documents are ranked in an ordered list according to their probabilities of relevance to the query. Besides, based on the within document term frequency and the query term frequency of the given search term, the search term is assigned with a weight using the weighting function in BM25.

$$W = \frac{(k_1+1)*tf}{K+tf} * \log\frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} * \frac{(k_3+1)*qtf}{k_3+qtf} \oplus k_2 * nq * \frac{(avdl-dl)}{(avdl+dl)} \quad (37)$$

where *N* is the number of indexed documents in the collection, *n* represents the number of documents which contain a specific term. *R* is the number of documents that are known to be relevant to a specific topic, and *r* is the number of relevant documents containing the specific term.The *tf* is the within-document term frequency, and *qtf* is within-query term frequency. The notion of *dl* is the length of the document, the *avdl* is the average document length, and the *nq* is the number of query terms. The k_{iS} are tuning constants. These values of the constants are empirically determined depending on the selected database and on the nature of the given queries. *K* equals to $k_1*((1-b)+b*dl/avdl)$, and the $\oplus$ indicates that, rather than for each term, the following component is added only once per document.

The workflow of the Figure 4 retrieval system is presented as follows. There will be two rounds of queries, namely the first or initial round and the feedback round, performed on each topic query. In the first round, the system draws the system input of the original full query topic in text into the topic and term processing module. Then the module will load the contents of the topic, including the indexes and other statistical information. After that, the lower level query engine will receive the terms and associated information from the multi-level indexing model. Meanwhile, the re-ranking model will receive the retrieved results at this moment. The Okapi BSS will work to generate a term-based list and a document-based list of ranked documents results and to return to the re-ranking module. The re-ranking module will further the analysis on the two lists by re-distributing the weights of the terms and processing a new ranking list. According to the BM25 weighting function, the first *k* results will become the feedback, which allowing the flow of the initial round of retrieval to be finished. The result of this initial round will then go to the feedback processing module, and the final result comes from the feedback round of the information retrieval process.

In the experiments conducted of this thesis, I set b = 0.75 and k_1 = 1.2 during the calculation of the weighting functions, since these two values are usually applied as default parameter settings in most BM25 applications [11].

4.2 Data Sets

The experiments of the thesis are conducted on six representative TREC collections of data. The six selected datasets are WT2G (topic 401-450), Disk1&2 (topic 51-200), Disk4&5 (topic 401-450), WT10G'00 (topic 451-500), WT10G'01 (topic 501-550), and Blog06 (topic 851-950). The statistics of the selected data collections are shown in Table 1. These collections are varied in topics, sizes and contents, which serve to carry out a thorough evaluation of the proposed model and algorithms.

Collection	**TREC**	**Topics**	**#Docs**
WT2G	TREC8	401-450	247,491
Disk4&5	TREC8	401-450	528,155
Disk1&2	TREC1-3	51-200	741,858
Web10G	TREC9	451-500	1,692,096
Web10G	TREC10	501-550	1,692,096
Blogs06	TREC15-17	851-950, 1000-1050	3,215,171

Table1. TREC Collections

The WT2G collection is a 2G size general Web crawl used by the TREC 1999 Web track. It consists of 247,491 general Web documents (TREC'99 Web track).

The Disk 4&5 contains 528,155 newswire articles from various sources, such as the Financial Times (FT) and the Federal Register (FR), which are usually considered as high-quality text data with little noise (TREC'97-99 Ad hoc track).

The Disk1&2 collection contains 741,856 news articles from varied sources, such as Wall Street Journal and Associated press newswire.

The WT10G collection is a medium size crawl of 1,692,096 Web documents (TREC'00-01 Web track), containing 10 Gigabytes of uncompressed data. It was used in the TREC9 and TREC 10 Web track.

The Blog06 collection consists of 3,215,171 blog feeds collected over an 11 week period from December 2005 to February 2006.

<top>
<num> Number: 725
<title> Low white blood cell count
<desc> Description:
What would cause a lowered white blood cell count?
<narr> Narrative:
A relevant document will describe a condition or disease that causes a lowered white blood cell count. Lowered white blood cell counts caused by HIV infection, bone marrow failure and chemotherapy are relevant. A low count caused by a treatment or medication would also be relevant.
</top>

Figure 3. An Example of a Standard TREC topic

A standard TREC topic usually includes three topic fields which are the title, a description, and the narrative. The above Figure 3 gives an example of a standard TREC topic. In the experiments, only the short title topic field which contains valuable keywords that are related to the topic are applied. The title in most of the cases is a generalization of the users queries, and is more practical to use in the experiments.

The evaluation of the proposed new method is adopted with a ten-fold cross-validation for each data collection. It means that the each test topic in the collection is randomly split and assigned to ten equal subsets, where in each fold, nine out of ten subsets of the topics are applied for training, and the other subset is used for testing. On top of that, the overall retrieval performance is generated by taking an average on all ten test subsets of topics.

Following the official TREC settings [59], for all the test collections used in the experiments, only the permalinks, which consist of blog posts and their associated comments, are indexed. Also, Porter's English stemmer is applied to each term, and the Standard English stop words are removed. Each topic contains three topic fields, namely title, description and narrative, and the study only uses the title topic field that contains limited keywords related to

the topic. The title-only queries are usually short and are realistic snapshots of the real users' queries in practice.

Altogether, the data collections helped the thesis to prove that the proposed model give a comparatively better performance than the other single baseline models in general.

4.3 Gold Standard

A gold standard for the tasks judgment is the correctness responses to a query judged by authoritative and comprehensive knowledge. The gold standard of relevance and aspects on the relative term frequency and BM25 topics that I am using in this thesis to judge the documents is officially provided by TREC. According to the TREC, there are usually at least three reviewers to review the work. These judges are mainly recruited from the TREC participants coming from different field-related institutions, and other academic units or research centers. In most cases, a certain education level showing their significant professionalism in domain knowledge is required, which is usually in a form of a Doctor of Philosophy in the field of science.

The procedure of reviewing the correctness of the work is usually provided to the judges with a set of detailed instructions. The judges are usually given the work to review the topic questions and to identify some topic related key concepts. Then the judges would sit together and discuss

the relevant paragraphs and pick out the minimum complete and correct excerpts from the big pool.

4.4 Evaluation Metrics

In this thesis, all experiments are adopted with the TREC official evaluation measures, namely the topical mean average precision (MAP) on Blog06 [59], and the MAP on the other collections to evaluate the experimental results. In order to emphasize on the top retrieved entities, P@10 will be highlighted as evaluation measures to show the performance in the conducted experiments. All statistical tests are based on two-tailed Wilcoxon Matched-pairs Signed-rank test, with the application of a significance level of 0.05.

Chapter 5 Experimental Results

In this chapter, I present all the experiments conducted for the thesis to illustrate the effectiveness of the newly proposed influence functions which are integrated into BM25 model. The results obtained from series of experiments are based on the six representative TREC collections of data, are reported. These results serve to evaluate the effectiveness of the introduced model and algorithms.

	WT2G	
	MAP	P@10
BM25	.2585	.4320
Linear	.2840* (+9.8646%)	.4480 (+3.7037%)
Quadratic	.2852* (+10.3288%)	.4580* (+6.0185%)

Cube	.2865* (+10.8317%)	.4700* (+8.7963%)

Table 2. Comparison of BM25-RTF on different IFs and BM25 on WT2G, wrt MAP and P@10 where '' indicates a significant improvement over BM25 (Wilcoxon Matched-pairs Signed-rank test with $p < 0.05$).*

	Disk 4&5	
	MAP	P@10
BM25	.2409	.4729
Linear	.2464* (+2.2831%)	.4860* (+2.7701%)
Quadratic	.2437 (+1.1623%)	.4760 (+0.6555%)
Cube	.2435 (+1.0793%)	.4740 (+0.2326%)

Table 3. Comparison of BM25-RTF on different IFs and BM25 on Disk 4&5, wrt MAP and P@10 where '' indicates a significant improvement over BM25 (Wilcoxon Matched-pairs Signed-rank test with $p < 0.05$).*

	Disk 1&2

	MAP	P@10
BM25	.2127	.4600
Linear	.2195 (+3.1970%)	.4707 (+2.3261%)
Quadratic	.2207* (+3.7612%)	.4827* (+4.9348%)
Cube	.2215* (+4.1373%)	.4920* (+6.9565%)

Table 4. Comparison of BM25-RTF on different IFs and BM25 on Disk12, wrt MAP and P@10 where '' indicates a significant improvement over BM25 (Wilcoxon Matched-pairs Signed-rank test with $p < 0.05$).*

	WT10G'00	
	MAP	P@10
BM25	.1873	.2458
Linear	.1896 (+1.2280%)	.2604* (+5.9398%)
Quadratic	.1921 (+2.5627%)	.2708* (+10.1709%)
Cube	.1910	.2563*

	(+1.9754%)	(+4.2718%)

Table 5. Comparison of BM25-RTF on different IFs and BM25 on WT10G'00, wrt MAP and P@10 where '' indicates a significant improvement over BM25 (Wilcoxon Matched-pairs Signed-rank test with $p < 0.05$).*

	WT10G'01	
	MAP	P@10
BM25	.1887	.3460
Linear	.1982* (+5.0344%)	.3558 (+2.8324%)
Quadratic	.1987* (+5.2994%)	.3600* (+4.0462%)
Cube	.1962* (+3.9746%)	.3640* (+5.2023%)

Table 6. Comparison of BM25-RTF on different IFs and BM25 on WT10G'01, wrt MAP and P@10 where '' indicates a significant improvement over BM25 (Wilcoxon Matched-pairs Signed-rank test with $p < 0.05$).*

	Blogs06	
	MAP	P@10

BM25	.2879	.6053
Linear	.2914 (+1.2157%)	.5900 (-2.5277%)
Quadratic	.2936* (+1.9799%)	.5840 (-3.5189%)
Cube	.2935* (+1.9451%)	.5850 (-3.3537%)

Table 7. Comparison of BM25-RTF on different IFs and BM25 on Blogs06, wrt MAP and P@10 where '' indicates a significant improvement over BM25 (Wilcoxon Matched-pairs Signed-rank test with $p < 0.05$).*

Table 2-7 show the performance comparison between BM25-RTF with the application of each of the three newly proposed influence functions, and traditional probabilistic weighting model BM25 on six selected datasets over MAP and P@10. All the statistical tests are run based on the Wilcoxon Matched-pairs Signed-rank test.

With regard to MAP, the test results indicated that, in general, BM25-RTF could provide a significantly better performance than the BM25 could do on four out of the six collections. More specifically, the data collections showing better results are the WT2G, Disk1&2, WT10G'01, and Blogs06. For the other two datasets, namely the Disk4&5 and WT10G'00, the newly proposed

method, BM25-RTF still outperforms the classic BM25 weighting function, however, it does not show such significant improvements as the experiments run on the other four data collections.

With regard to P@10, as the numbers shown in the Table 2-7 above, the BM25-RTF has also generated significantly better performance results than the traditional BM25 on four of the six TREC data collections. The four datasets are the WT2G, Disk1&2, WT10G'00, and WT10G'01. On the other two of the six collections, the Disk4&5 and the Blogs06, our BM25-RTF still outperforms the BM25 with a slight difference.

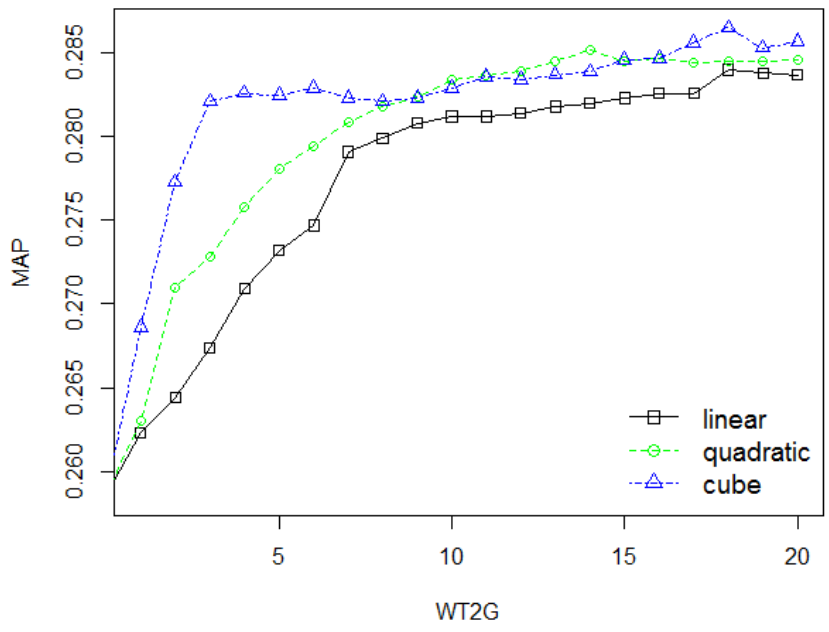

Figure 3. Influence Functions Performance on WT2G, MAP

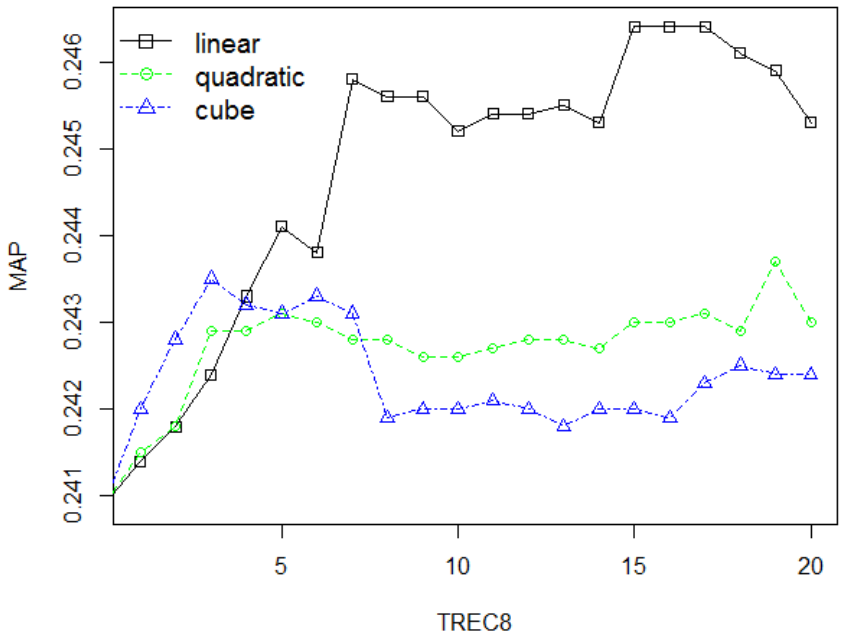

Figure 4. Influence Functions Performance on Disk4&5, MAP

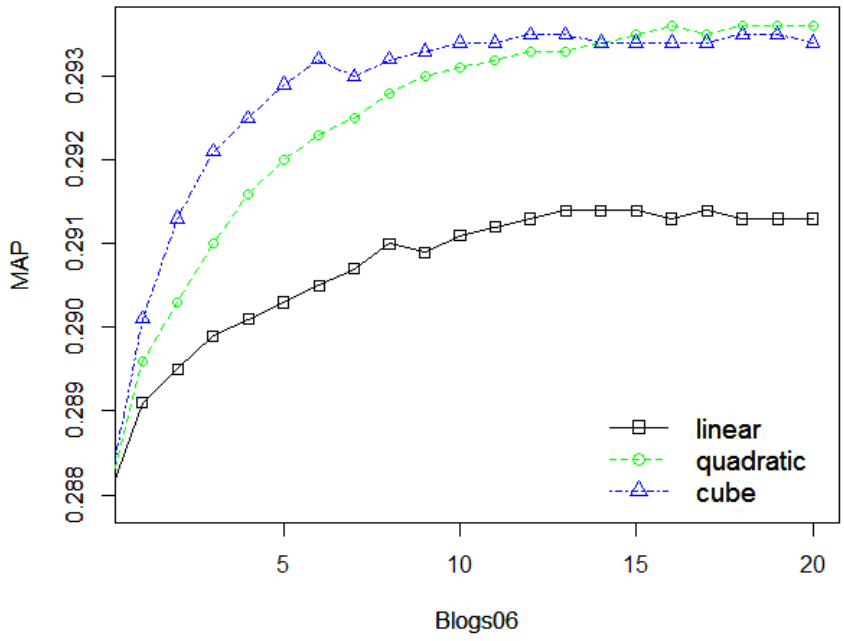

Figure 5. Influence Functions Performance on Blogs06, MAP

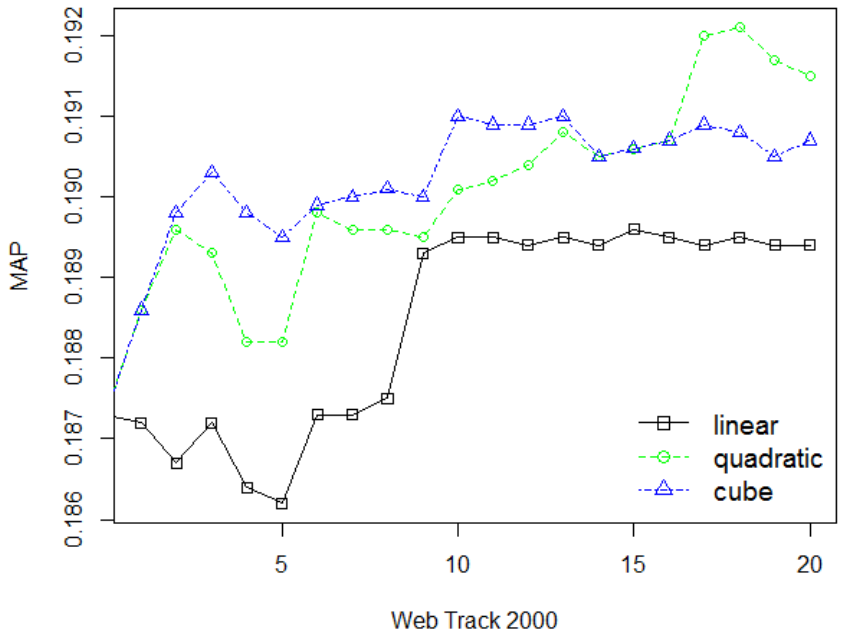

Figure 6. Influence Functions Performance on Web Track 2000, MAP

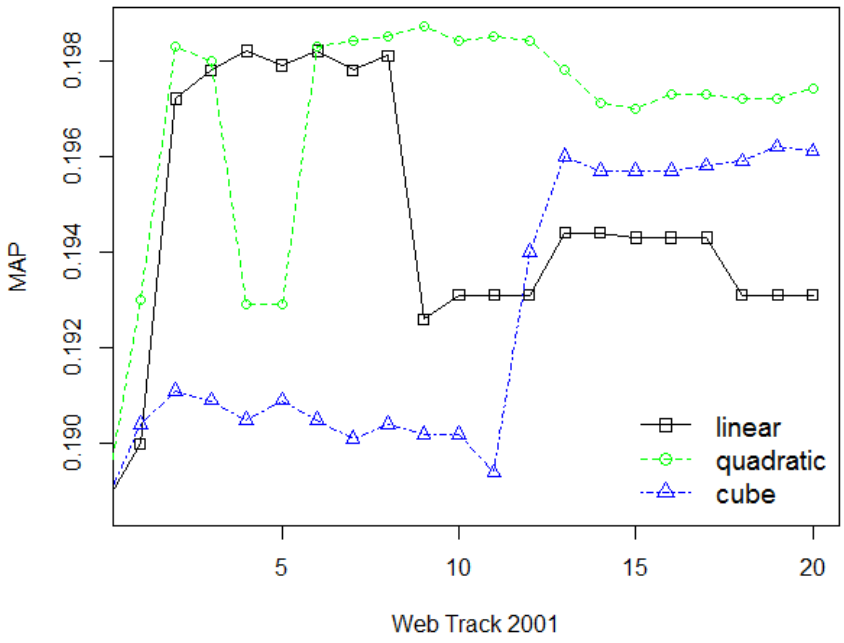

Figure 7. Influence Functions Performance on Web Track 2001, MAP

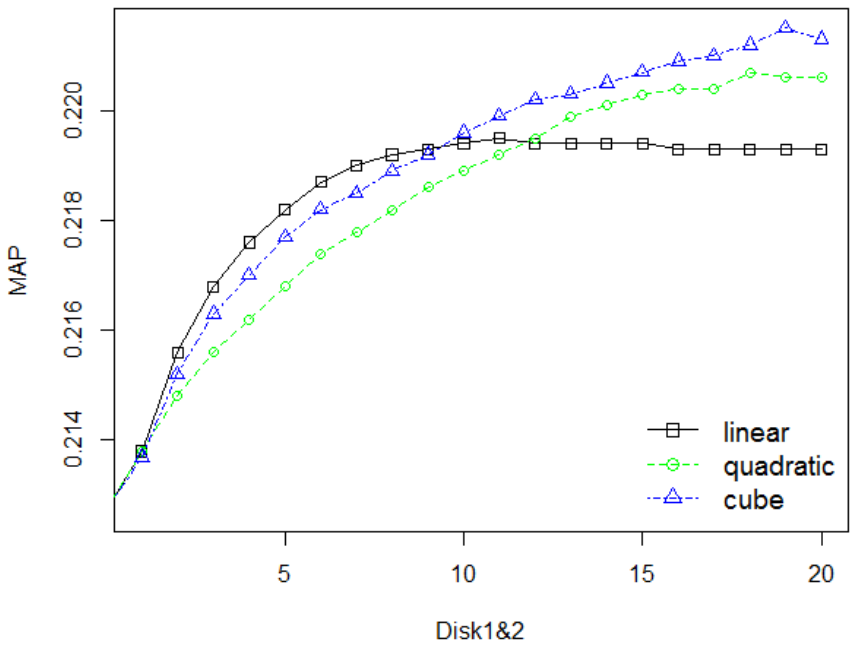

Figure 8. Influence Functions Performance on Disk 1&2, MAP

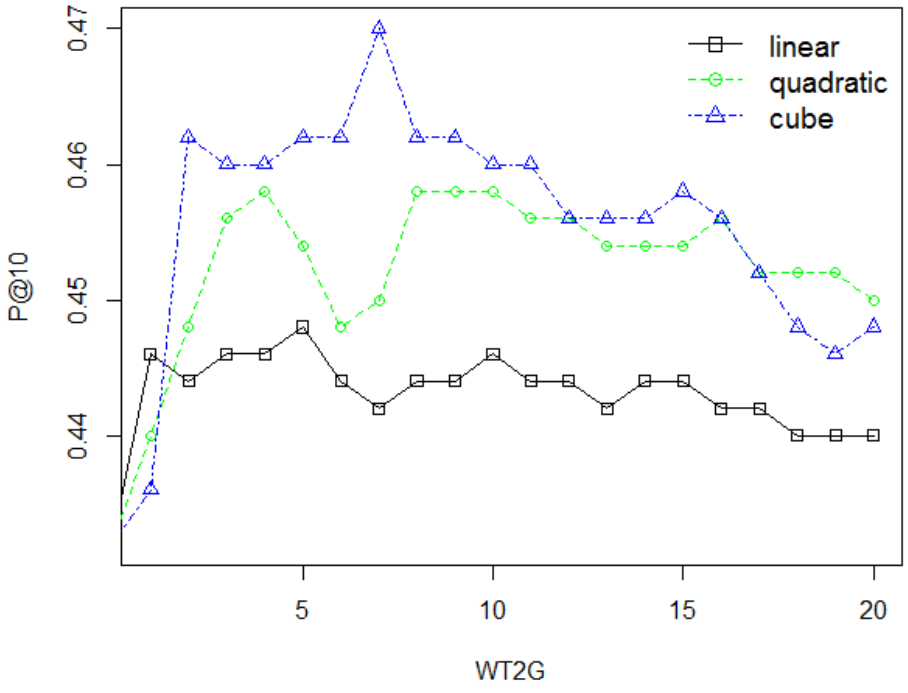

Figure 9. Influence Functions Performance on WT2G, P@10

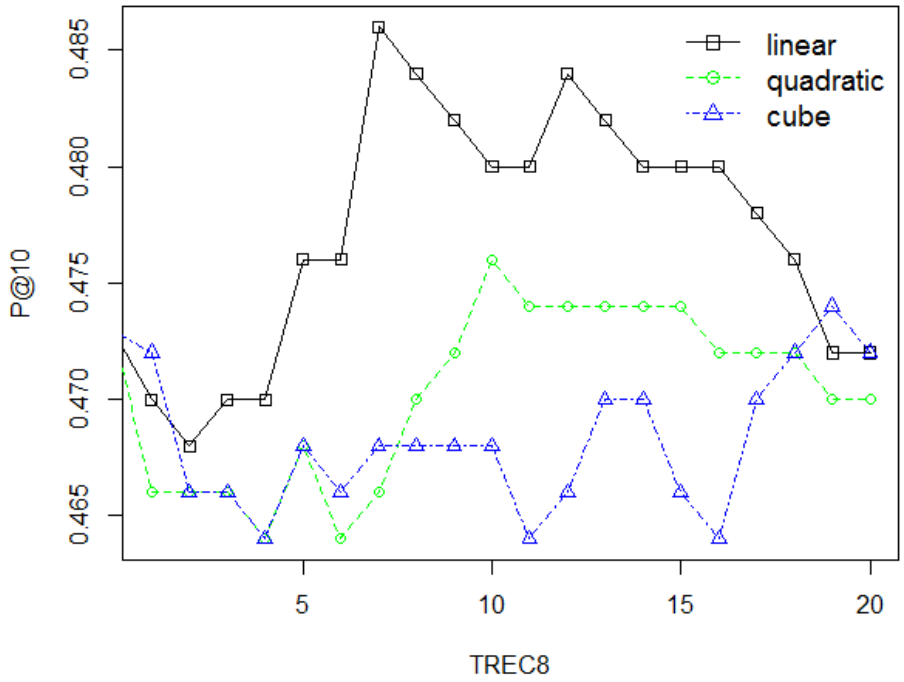

Figure 10. Influence Functions Performance on TREC8, P@10

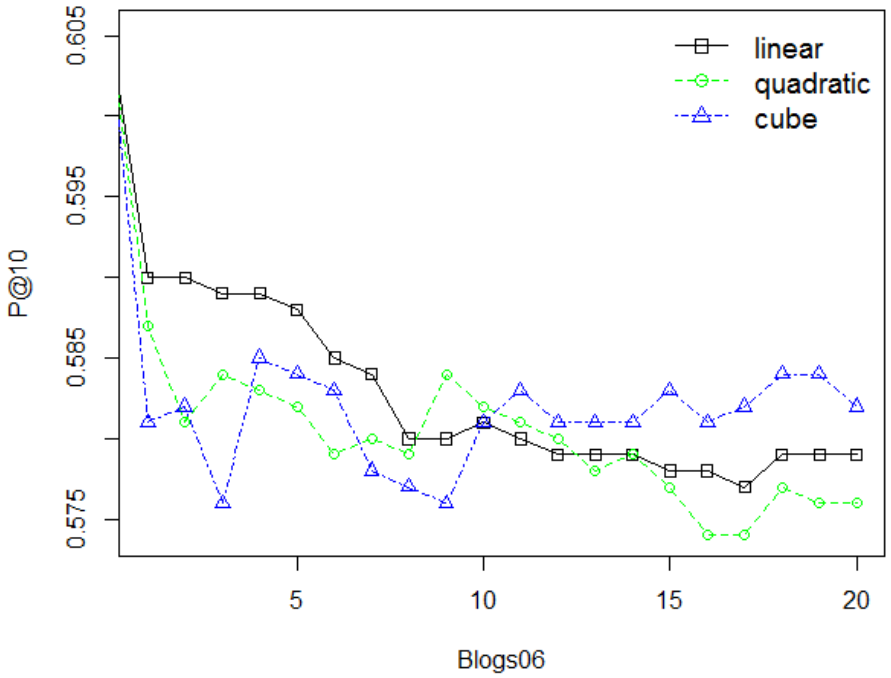

Figure 11. Influence Functions Performance on Blogs06, P@10

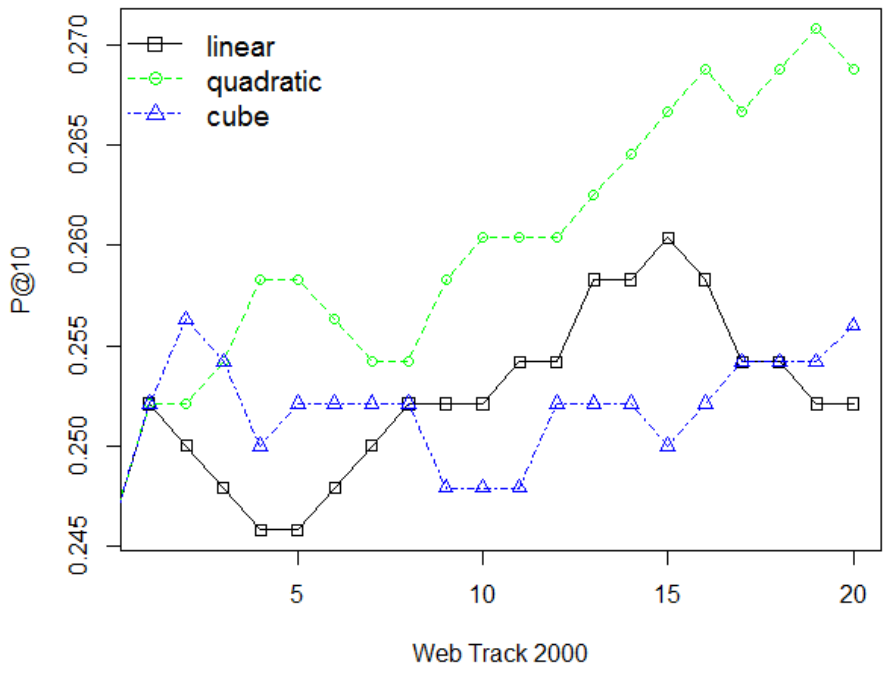

Figure 12. Influence Functions Performance on Web Track 2000, P@10

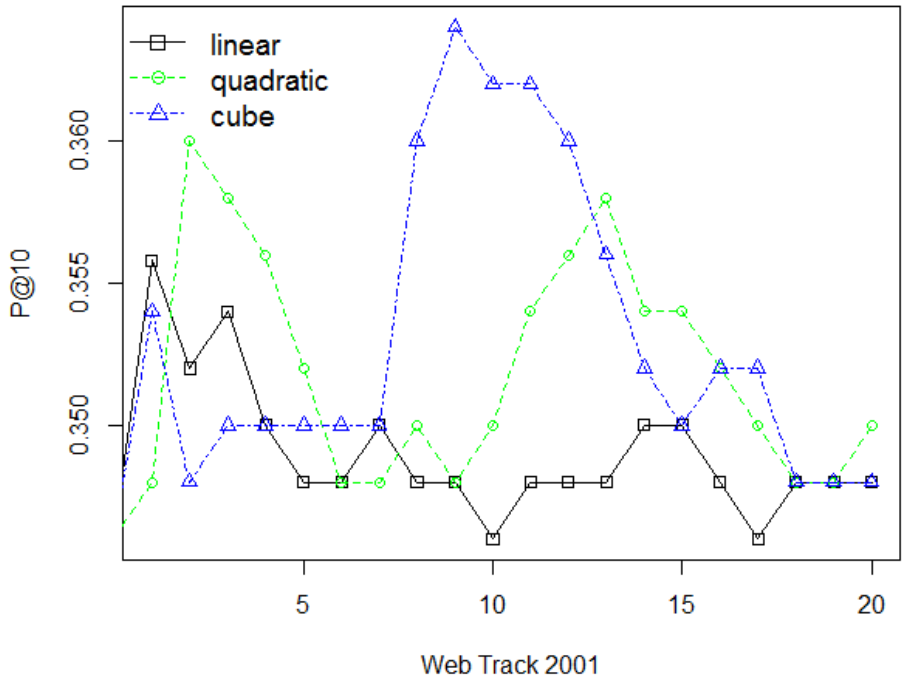

Figure 13. Influence Functions Performance on Web Track 2001, P@10

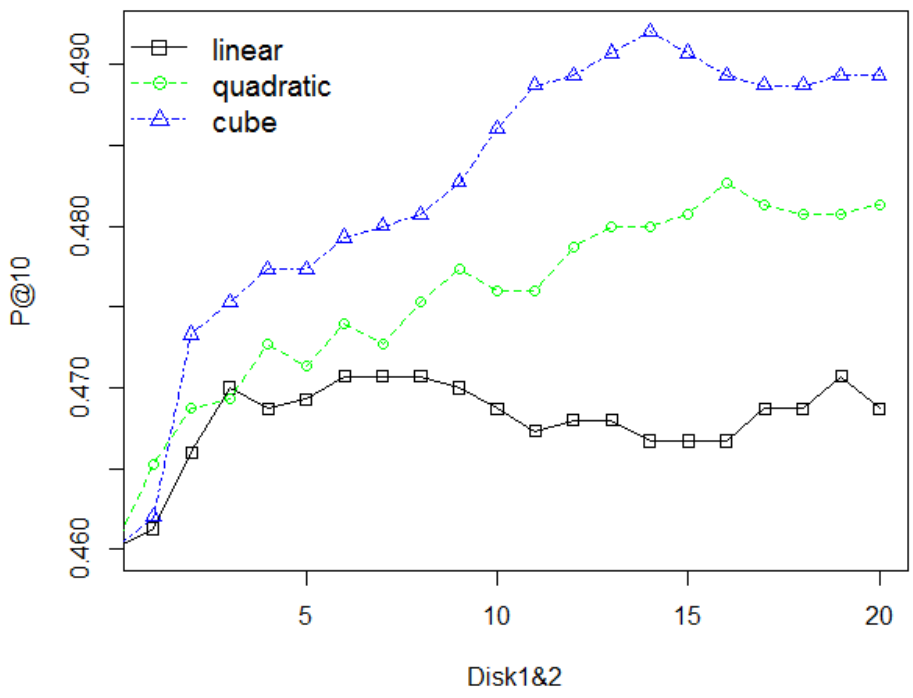

Figure 14. Influence Functions Performance on Disk 1&2, P@10

Chapter 6 Analyses and Discussions

The experimental results will be further studied and analyzed in this chapter to see how the proposed three influence functions with high relative term frequency could serve to present better results than the single classic BM25 function.

The ability of judging the effectiveness of a proposed new model is always a challenging task. This thesis managed to evaluate, compare and contrast the experimental results with the help of the evaluation metrics which our lab has. The evaluation metrics span over the past few years in both the text based track data and the Blog track data. The criteria for measuring the performance of a single run of given information retrieval is analyzed with different techniques. The most important measurement in the analyses is the mean average precision (MAP), which refers to the mean of the average precision scores taken from the sum of the average precision score calculated after the system completes extraction and retrieval for each relevant document. At the same time, the P@n value which is the number of relevant documents retrieved in the rank of top n documents will be included

for analysis. For the discussions, the thesis is interested in n=5, n=10 or n=20, thus, the analyses were done on how many of the top five, ten or twenty documents retrieved are relevant to the query given by the users indeed.

To illustrate the performance differences graphically, the experimental results are plotted in the Figures 15 to 16 below respectively.

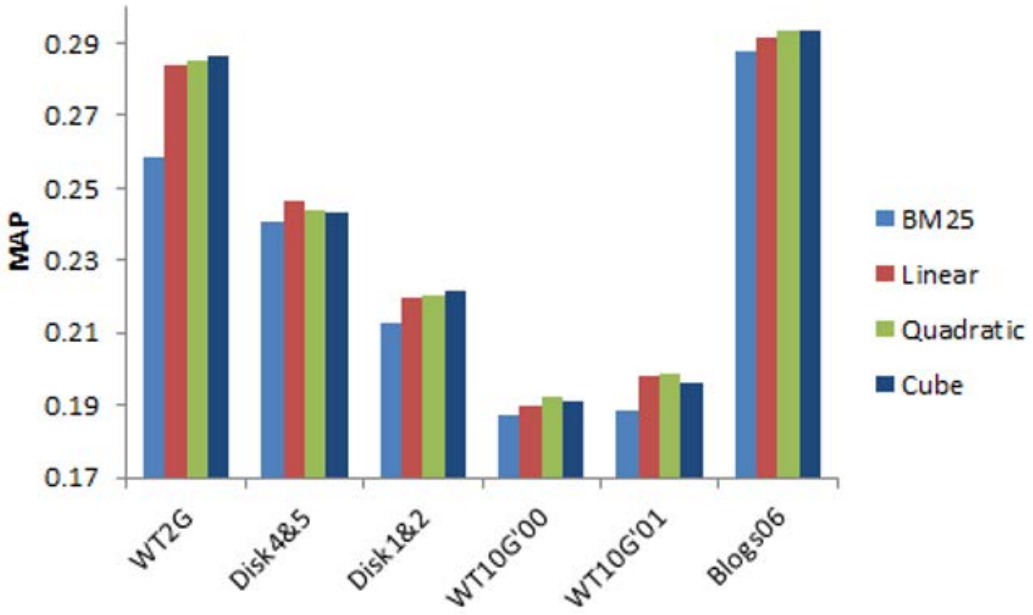

Figure 15. Performance Comparison on MAP

In the figure 15 above, the performance of each evaluation approach is shown in different colors. We could see in general, the substitution of relative term frequency into the Linear, Quadratic and Cube functions performs better than the single BM25 model. More specifically, the Cube function outperforms or performs as well as the Linear and Quadratic

function in the datasets of WT2G, Disk 1&2, and Blogs06. Whereas, the Linear function with the high relative term frequency performs better in Disk 4&5, and the Quadratic function gives an overall stable and robust test results.

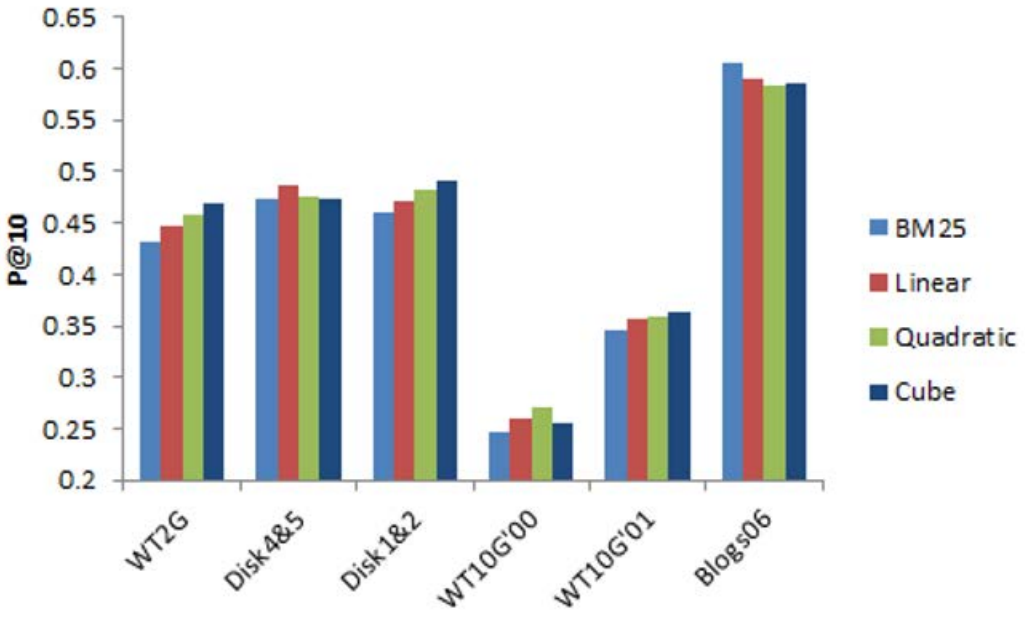

Figure 16. Performance Comparison on P@10

Similar to the diagram above, Figure 16 above also clearly indicates that BM25-RTF model generally outperforms the BM25 on both MAP and P@10. Except for the Blog Track dataset Blogs06, where BM25 gives a slightly better result in the Blogs06, the Linear, Quadratic and Cube functions are more effective in improving the experimental results. The improvements are significant on most data collections. As the Table 2-7 have shown, the BM25-RTF integrated Quadratic and Cube influence functions generally achieve

better performance results than the BM25-RTF integrated Linear influence function, on both MAP and P@10. It turns out that both the Quadratic and the Cube influence functions represent smaller contributions with the application of the relative term frequency than Linear function based on the experimental results we have now.

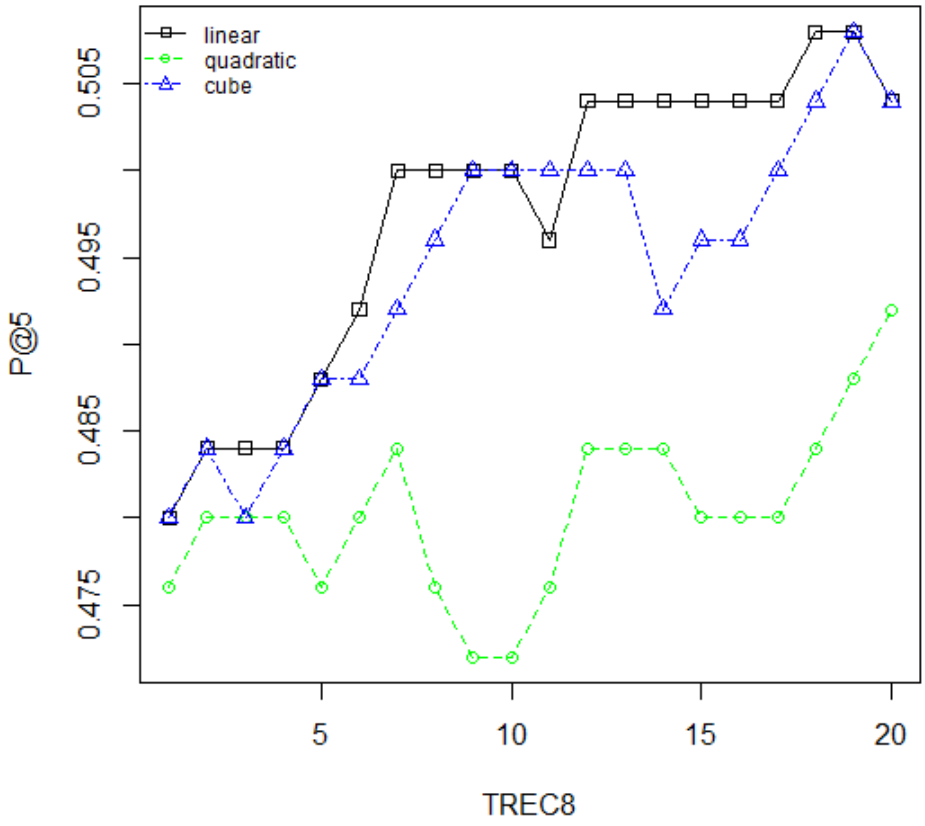

Figure 17. P@5 results on TREC8

The Figure 17 gives the experimental results on TREC8 with the application of P@5 analysis. In general, the high relative term frequency integrated linear function performs well and shows a stably increasing

tendency. Comparatively, the quadratic function stays in a range of 0.470 to 0.490, but it drops significantly and rises significantly in the value of 6 and 12 respectively.

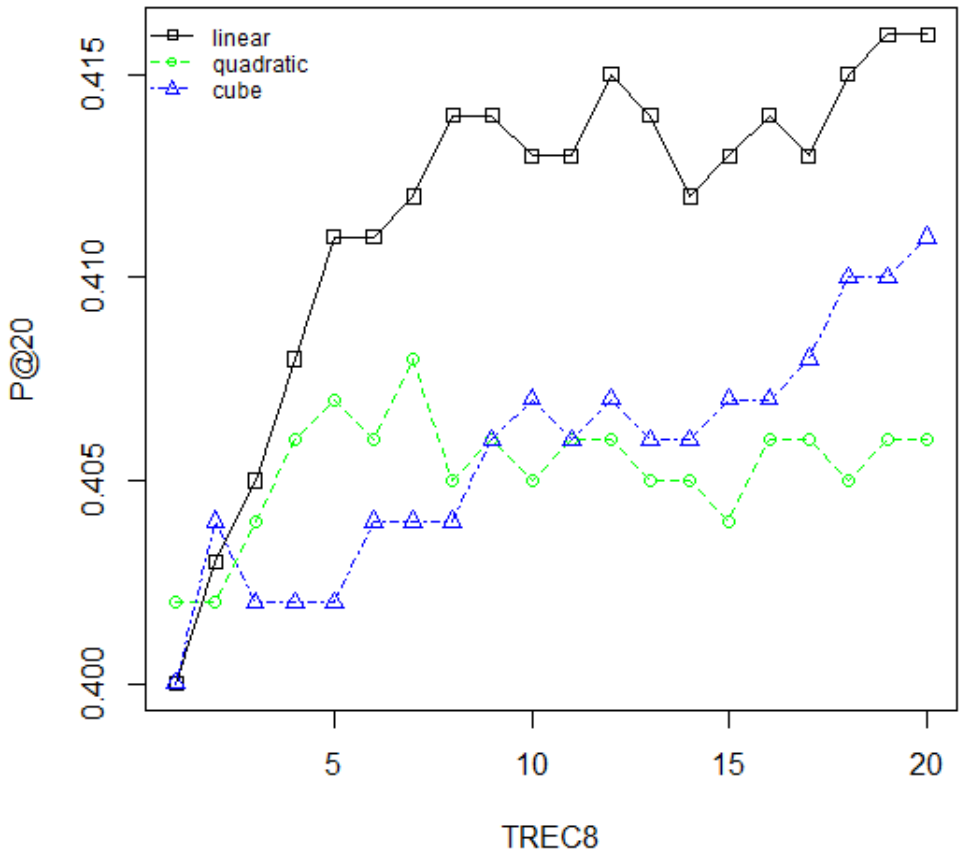

Figure 18. P@20 results on TREC8

The P@20 of Linear Function climbs as high as the P@5 results as the TREC8 data goes from 1 to 20. However, when we look closely at the P@20 values, the Quadratic influence function is comparatively more stable, and goes into a similar pattern as the Cube influence function.

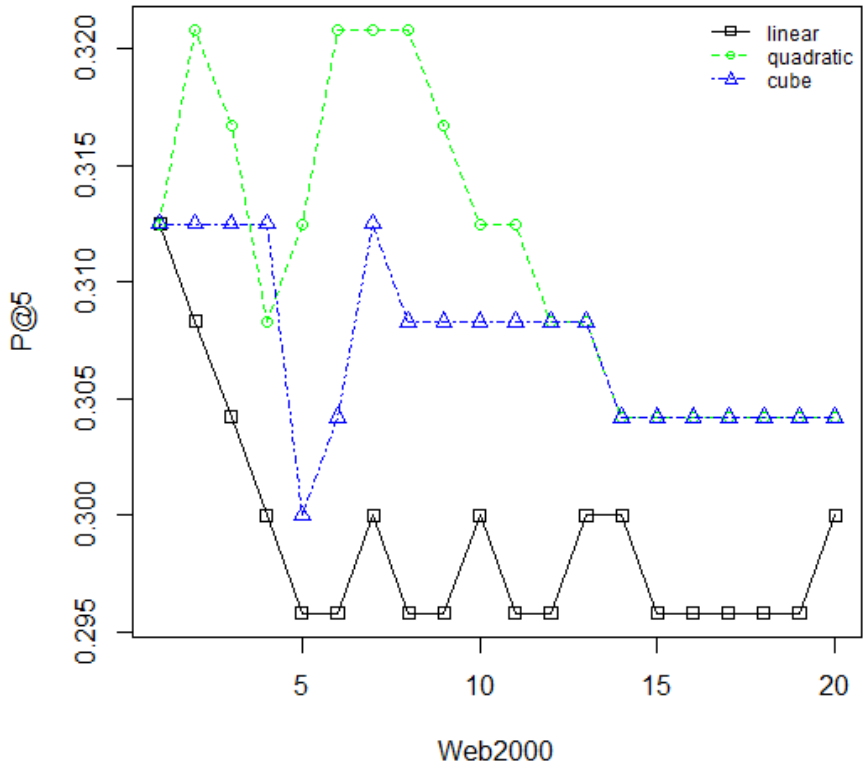

Figure 19. *P@5 results on Web2000*

We can see that the experimental results here on the Web2000 are in a very different shape. All three influence functions are in a decreasing tendency as the values grow from 1 to 20. For the relative term frequency adopted linear function, P@5 drops significantly from around 0.3125 at 1 to around 0.2951 at 5. And the P@5 remains in the range of 0.2951 to 0.3000 by then. However, both the relative term frequency integrated Quadratic and Cube functions experience a sharp raise after a similar big drop, but they both keep through a smooth value line at 0.305.

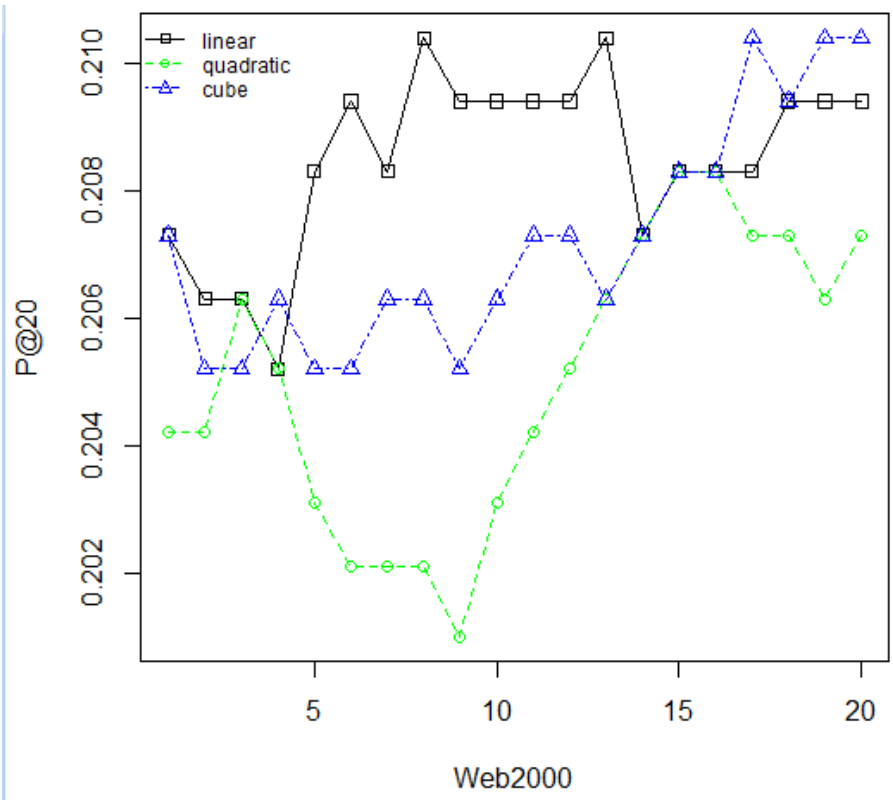

Figure 20. *P@20 results on Web2000*

All three proposed functions perform pretty well in P@20, Web2000 with a generally increasing tendency. Based on the results we have above, we can see the expected big drop of the Quadratic function and a steep returning line from 9 to 15. All other P@5, P@20 results will be presented below, and we could see on different datasets, the relative term frequency integrated Linear, Quadratic and Cube functions reveal different comparative advantages.

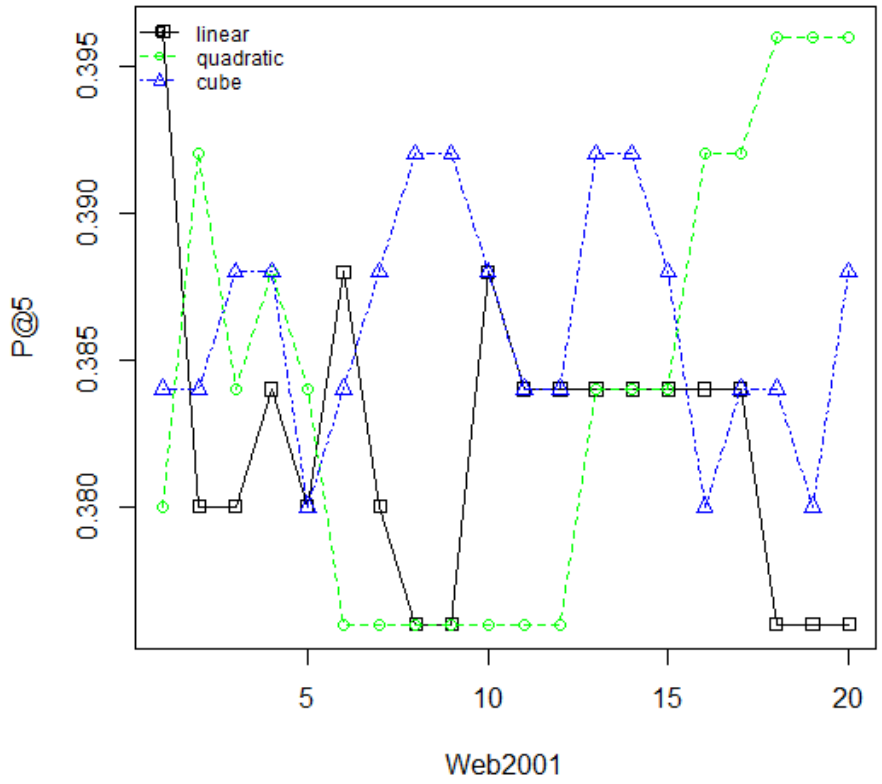

Figure 21. *P@5 results on Web2001*

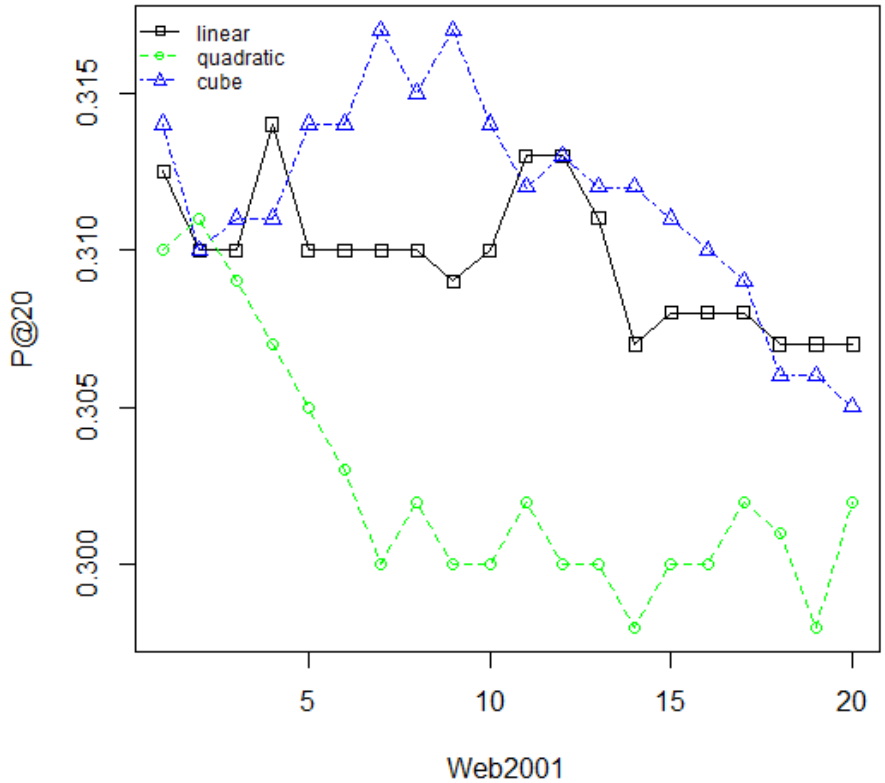

Figure 22. *P@20 results on Web2001*

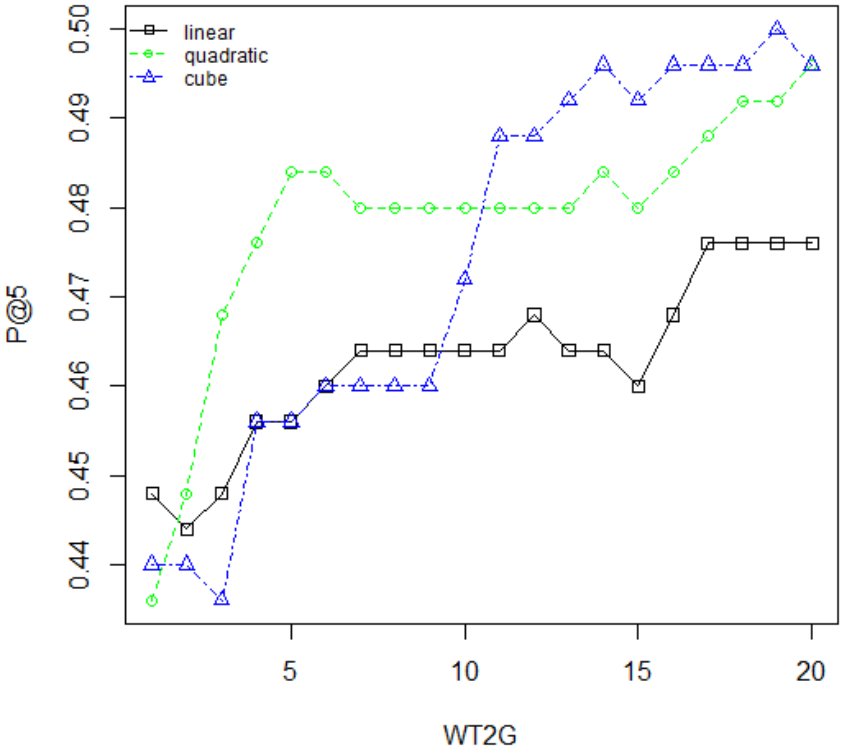

Figure 23. *P@5 results on WT2G*

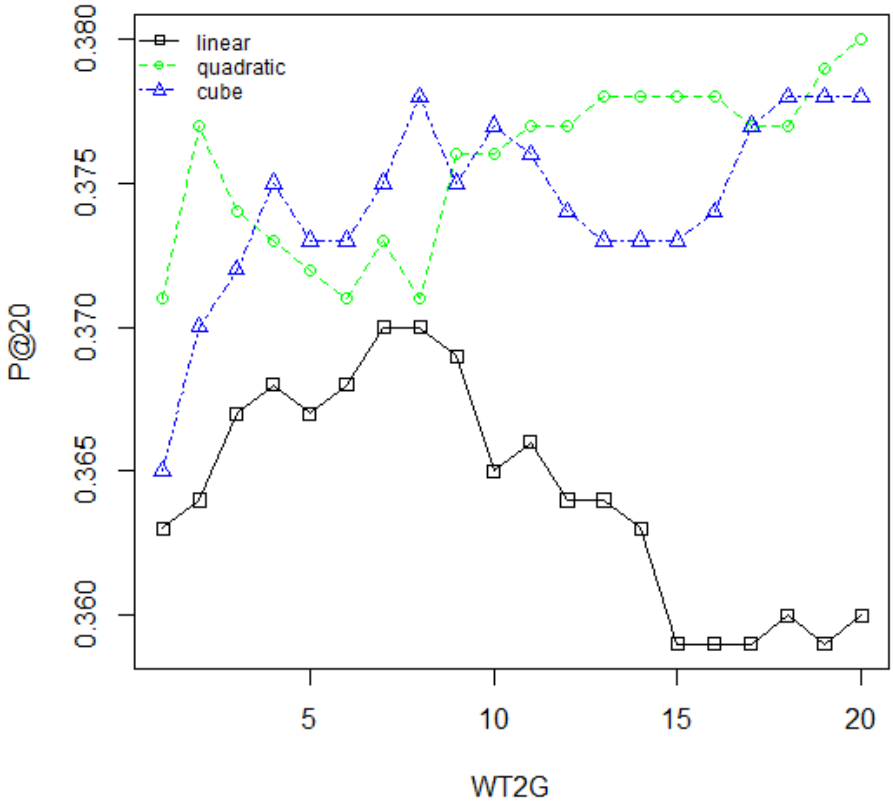

Figure 24. *P@20 results on WT2G*

As what we have seen above, the Figures 3 to 24 show how the parameter β in the influence functions impacts the retrieval performance. As discussed in Chapter 2, the BM25-RTF becomes BM25 when β = 0, and as β increases, the BM25-RTF takes more relative term frequency into account. It is shown that the performances of BM25-RTF on all MAP, P@5, P@10 and P@20 increase at first when β increments from 0 over most data collections. This indicates that the proposed term rewarding technique does boost BM25-RTF's performance significantly. On contrary, as β keeps incrementing, the performances start to decrease due to the overvaluing of the relative term frequency. Here the two below diagrams are the experimental results of tuning beta of b=0.3, which linearly combines BM25 with the Quadratic function. The thesis proposes only the results of WT2G and Blogs06 here.

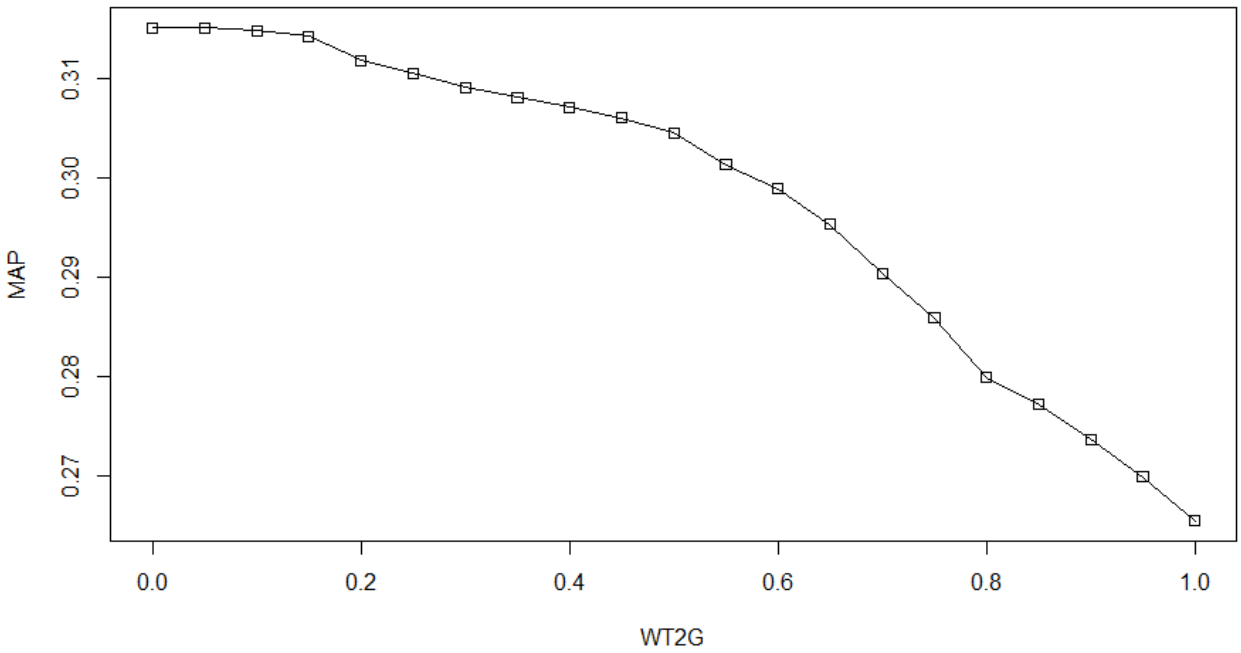

Figure 25. Tuning Beta on WT2G

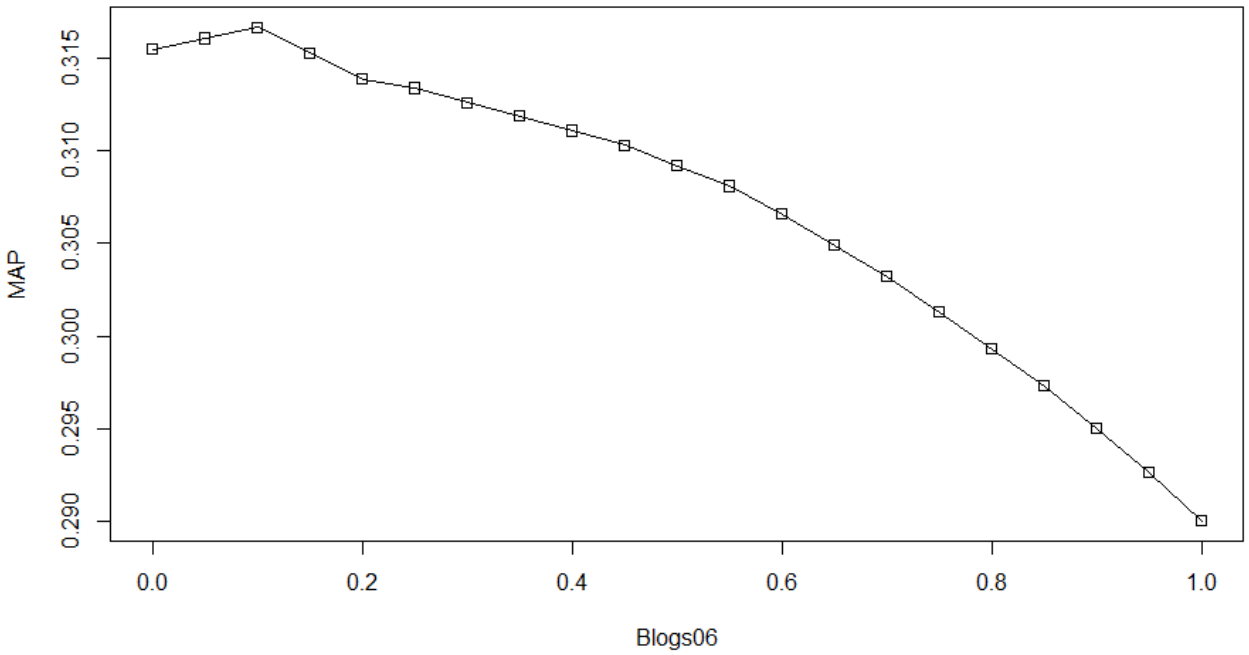

Figure 25. Tuning Beta on Blogs06

Chapter 7 Conclusion & Future Work

7.1 Conclusions

In this thesis, I propose a BM25-RTF model to reward terms according to their relative frequencies in a document. The focus is to suggest that a term with high relative frequency within a document is more representative and relevant in the document characterization and ranking. Based on the research and analysis work, I propose and present three influence functions to directly replace and integrate the relative term frequency information into the traditional BM25 weighting function. The experiments run for the proposed approaches and framework has placed their capacity of generating higher accuracy of retrieval at specific levels. Meanwhile, the experimental results also show that the new model BM25-RTF, which is integrated with relative term frequency information, significantly outperforms BM25 on MAP and P@10 on most of the six representative data collections. It is a novel approach to combine the concept of relative term frequency with fundamental weighting functions in probabilistic information retrieval

systems in order to achieve better performance for retrieval results. The framework is accurate and applicable according to specified requirements.

7.2 Future Work

Although the results obtained from the experiments are ideal to some extent, the findings in this thesis still raise several problems which need to be explored and investigated further in the near future.

1) This relative term frequency concept replacement in the BM25 classic function has proved to be successful in the study, however, due to the strict timing issue, the results of a linear combine between the proposed influence functions and the BM25 function still remain unexplored. With the merge of different functions, the values could be computed more carefully, and will thus rank the documents more effectively and efficiently.
2) From the conducted experiments, I noticed that based on the fact that both Quadratic and Cube influence functions represent smaller contributions of relative term frequency than linear function based on the experimental results we have now, in the future, we would need to study more functions to investigate this tendency.

3) The relative term frequencies are more likely to be analyzed and evaluated under the scope of the relevant term weighting methods instead of considering them individually. I believe it is necessary to take in concepts like cross term, and bag of words.
4) In the future, further works should also apply BM25-RTF to more datasets to further investigate the effect of the proposed influence functions. On the other hand, researchers should explore other ways to represent the influence of the relative term frequency, and the integration of influence functions with other IR models. It is acceptable to study the retrieval process by the applications of more advanced text mining and statistics methods to boost performance. And of course, more performance metrics should be examined.

Bibliography

[1] J. H. Paik. A novel tf-idf weighting scheme for effective ranking. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, pages 343{352, New York, NY, USA, 2013. ACM.

[2] S. Clinchant and E. Gaussier. Retrieval constraints and word frequency distributions a log-logistic model for IR. Inf. Retr., 14(1):5–25, Feb. 2011.

[3] H. Fang and C. Zhai. Diagnostic evaluation of information retrieval models. Trans. Inf. Syst., 29(2):1–42, April 2011.

[4] W. R. Greiff. A theory of term weighting based on exploratory data analysis. In SIGIR'98, pages 11–19.

[5] B. He and O. I. On setting the hyper-parameters of term frequency normalization for information retrieval. Trans. Inf. Syst., 25(3), July 2007.

[6] B. He and I. Ounis. A study of the dirichlet priors for term frequency normalisation. In SIGIR'05, pages 465–471.

[7] K. L. Kwok. A new method of weighting query terms for ad-hoc retrieval. In SIGIR'96, pages 187–195.

[8] Y. Lv and C. Zhai. Adaptive term frequency normalization for BM25. In CIKM'11, pages 1985–1988.

[9] Y. Lv and C. Zhai. A log-logistic model-based interpretation of TF normalization of BM25. In ECIR'12, pages 244–255.

[10] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In CIKM'11, pages 7–16.

[11] Harter S.P. A probabilistic approach to automatic keyword indexing. Journal of the American Society for Information Science 1975; 26:197-206 and 280-289.

[12] S. Robertson and M. Beaulieu. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. In TREC'99, pages 253–264.

[13] S. Robertson and M. Taylor. Simple BM25 extension to multiple weighted fields. In CIKM, pages 42–49. ACM, August 2004.

[14] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In SIGIR' 94, pages 232–241.

[15] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. Inf. Process. Manage., 24(5):513–523, August 1998.

[16] A. Singhal and M. Mitra. Pivoted document length normalization. In SIGIR'96, pages 21–29.

[17] Z. Ye and J. Huang. A simple term frequency transformation model for effective pseudo relevance feedback. In SIGIR'14, pages 323–332.

[18] Stanford. Introduction to Information Retrieval. In Online edition (c) 2009, Cambridge UP

http://nlp.stanford.edu/IR-book/pdf/12lmodel.pdf

[19] Beigbeder, M. and Mercier, A. (2005). An information retrieval model using the fuzzy proximity degree of term occurences. In Proceedings of the 2005 ACM Symposium on Applied Computing, pages 1018–1022. ACM New York, NY, USA

[20] Gao, J., Nie, J., Wu, G., and Cao, G. (2004). Dependence language model for information retrieval. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 170–177. ACM New York, NY, USA.

[21] Hawking, D. and Thistlewaite, P. (1995). Proximity operators - So near and yet so far. In Proceedings of the 4th Text Retrieval Conference, pages 131–143.

[22] Song, F. and Croft, W. (1999). A general language model for information retrieval. In Proceedings of the eighth International Conference on Information and Knowledge Management, pages 316–321. ACM New York, NY, USA.

[23] Srikanth, M. and Srihari, R. (2002). Biterm language models for document retrieval. In Proceedings of the 25th Annual International

ACM SIGIR Conference on Research and Development in Information Retrieval, pages 425–426. ACM New York, NY, USA.

[24] Ahmed, F. and Nurnberger, A. (2009). Evaluation of n-gram conflation approaches for Arabic text retrieval. Journal of the American Society for Information Science and Technology, 60(7):1448–1465.

[25] Mayfield, J. and McNamee, P. (2003). Single n-gram stemming. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 415–416. ACM New York, NY, USA.

[26] Broschart, A. and Schenkel, R. (2008). Proximity-aware scoring for XML retrieval. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 845–846. ACM New York, NY, USA.

[27] B¨uttcher, S., Clarke, C., and Lushman, B. (2006). Term proximity scoring for adhoc retrieval on very large text collections. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 621–622. ACM New York, NY, USA.

[28] Jiashu, Z. (2015). Term Association Modelling in Information Retrieval. PhD. Dissertation, York University.

[29] Tao, T. and Zhai, C. (2007). An exploration of proximity measures in information retrieval. In Proceedings of the 30th Annual International

ACM SIGIR Conference on Research and Development in Information Retrieval, pages 295–302. ACM New York, NY, USA.

[30] Goker, A. and Davies, J. (2009). Information retrieval: searching in the 21st century. Wiley. Com

[31] Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). Modern information retrieval, volume 463. ACM press New York.

[32] Fuhr, N. (1992). Probabilistic models in information retrieval. The Computer Journal, 35(3):243–255.

[33] Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1996). Okapi at TREC-4. In Proceedings of the 4th Text Retrieval Conference, pages 73–97.

[34] Zhao, J., Huang, X., Ye, Z., and Zhu, J. (2009). York University at TREC 2009: Chemical track. In Proceedings of the 18th Text Retrieval Conference.

[35] Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 275–281. ACM.

[36] M. E. Maron and J. L. Kuhns. On Relevance, Probabilistic Indexing and Information Retrieval. J. ACM, 7(3):216–244, 1960. ISSN 0004-5411.

[37] William L. Miller. A Probabilistic Search Strategy for MEDLARS. Journal of Documentation, 27:254–266, 1971.

[38] M. Beaulieu, M. Gatford, Xiangji Huang, Stephen E. Robertson, S. Walker, and P. Williams. Okapi at TREC-5. In Proceedings of 5th Text REtrieval Conference, pages 143–166. NIST Special Publication, 1997.

[39] Xiangji Huang, F. Peng, D. Schuurmans, Nick Cercone, and Stephen E. Robertson. Applying Machine Learning to Text Segmentation for Information Retrieval. Information Retrieval Journal, 6(4):333–362, 2003.

[40] Stephen E. Robertson and Karen Sparck Jones. Relevance Weighting of Search Terms. Journal of the American Society for Information Science, 27:129–146, 1976.

[41] Stephen E. Robertson and Steve Walker. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 3-6 July 1994, Dublin, Ireland, pages 232–241, 1994.

[42] Stephen E. Robertson, M. E. Maron, and William S. Cooper. The Unified Probabilistic ModelforIR. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’82, pages 108–117, 1982.

[43] Fabio Crestani, Mounia Lalmas, C. J. van Rijsbergen, and Iain Campbell. “Is This Document Relevant? ... Probably”: A Survey of

Probabilistic Models in Information Retrieval. ACM Comput. Surv., 30(4):528–552, 1998.

[44] G. Salton. Automatic Information Organization and Retrieval. 1968.

[45] Steven Wartik. Information Retrieval. chapter Boolean Operations, pages 264–292. 1992. ISBN 0-13-463837-9.

[46] C. J. van Rijsbergen. A New Theoretical Framework for Information Retrieval. In Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’86, pages 194–200, 1986.

[47] C.J.van Rijsbergen. Probabilistic Retrieval Revisited. Comput.J.,35(3):291–298,1992.

[48] C. J. van Rijsbergen and Karen Sparck Jones. A Test for the Separation of Relevant and Nonrelevant Documents in Experimental Retrieval Collections. Journal of Documentation, 29:251–257, 1973.

[49] William S. Cooper. Some Inconsistencies and Misidentified Modeling Assumptions in Probabilistic Information Retrieval. ACM Trans. Inf. Syst., 13(1):100–111, 1995.

[50] Norbert Fuhr and Chris Buckley. A Probabilistic Learning Approach for Document Indexing. ACM Trans. Inf. Syst., 9(3):223–248, 1991.

[51] A. Bookstein and D.R. Swanson. Probabilistic Models for Automatic Indexing. Journal of the American Society for Information Science, 25:312–318, 1974.

[52] J. Yamron. Topic Detection and Tracking Segmentation Task. In Proceedings of the Topic Detection and Tracking Workshop, 1997.

[53] M. Zhong and X. Huang. Concept-based Biomedical Text Retrieval. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 6-11, 2006, Seattle, Washington, USA, pages 723–724. ACM, 2006.

[54] X. Huang, B. Hu, and H. Rohian. York University at TREC 2006: Genomics Track. In Proceedings of 15th Text REtrieval Conference, 2006.

[55] X. Huang, Y.R. Huang, and M. Wen. A Dual Index Model for Contextual Information Retrieval. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 2005, Salvador, Brazil, pages 613–614. ACM, 2005.

[56] X. Huang, Y.R. Huang, M. Wen, A. An, Y. Liu, and J. Poon. Applying Data Mining to Pseudo-Relevance Feedback for High Performance Text Retrieval. In Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China, pages 295–306, 2006.

[57] X. Huang, D. Sotoudeh-Hosseini, H. Rohian, and Xiangdong An. York University at TREC 2007: Genomics Track. In Proceedings of 16th Text REtrieval Conference, 2006.

[58] X. Huang, M. Wen, A. An, and Y.R. Huang. A platform for Okapi-based Contextual Information Retrieval. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 6-11, 2006, Seattle, Washington, USA, pages 728–728. ACM, 2006.

[59] Ounis, I., De Rijke, M., Macdonald, C., Mishne, G., and Soboroff, I. (2006b). Overview of the TREC-2006 Blog track. In Proceedings of the 15th Text Retrieval Conference.

[60] Peter Biebricher, Norbert Fuhr, Gerhard Lustig, Michael Schwantner, and Gerhard Knorz. The Automatic Indexing System AIR/PHYS – From Research to Application. In Proceedings of the 11st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '88, pages 333–342, 1988.

[61] Norbert Fuhr. Models for Retrieval with Probabilistic Indexing. Inf. Process. Manage., 25(1):55–72, 1989.

[62] Birger Hjørland and Frank Sejer Christensen. Work Tasks and Socio-cognitive Relevance: A Specific Example. J. Am. Soc. Inf. Sci. Technol., 53(11,).

[63] Hsiao, D. and Harary, F. (1970). A formal system for information retrieval from files. Communications of the ACM, 13(2):67–73.

[64] Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620.

[65] Gerard Salton, Edward A. Fox, and Harry Wu. Extended Boolean Information Retrieval. Commun. ACM, 26(11):1022–1036, 1983. ISSN 0001-0782.

[66] Stephen E. Robertson. The Probabilistic Character of Relevance. Inf. Process. Manage., 13(4):247–251, 1977.

[67] William Hersh, Aaron M. Cohen, and Phoebe Roberts. TREC 2006 Genomics Track Overview. In Proceedings of 15th Text REtrieval Conference. NIST Special Publication, 2006.

[68] Ye, Z. and Huang, X. J. "A Learning to Rank Approach for Quality-Aware Pseudo Relevance Feedback" (32 pages), Journal of the American Society for Information Science and Technology (JASIST). Wiley InterScience Publisher. In press, 2015. ISSN (Printed): 1532-2882 and ISSN (Online): 1532-2890.

[69] Hu, Q. and Huang, X. "Bringing Information Retrieval into Crowdsourcing", Proceedings of the 36th European Conference on Information Retrieval (ECIR'14), Amsterdam, Netherlands, April 13-16, 2014

[70] Zhao, J. and Huang, X. J. "An Enhanced Context-sensitive Proximity Model for Probabilistic Information Retrieval'', Proceedings of the 37th Annual International ACM SIGIR Conference on Research and

Development in Information Retrieval (SIGIR'14), Gold Coast, Australia. July 6-11, 2014

[71] Huang, X. J. Miao, J. and He, B. "High Performance Query Expansion Using Adaptive Co-training", Information Processing & Management: An International Journal (IPM). ELSEVIER Publisher. 49:441-453, 2013.

[72] Zhao, J., Huang, X. J. and Hu, T. "BPLT+: A Bayesian-based Personalized Recommendation Model for Health Care'' (18 pages), to appear in BMC Genomics. August 2013. ISSN: 1471-2164.

[73] Daoud, M. and Huang, X. J. "Modelling Geographic, Temporal and Proximity Contexts for Improving Geo-Temporal Search", Journal of the American Society for Information Science and Technology (JASIST). Wiley InterScience Publisher, 64(1):190-212, 2013. ISSN (Printed): 1532-2882 and ISSN (Online): 1532-2890.

[74] Daoud, M. and Huang, X. J. "Mining Query-Driven Contexts for Geographic and Temporal Search" (21 pages), accepted by International Journal of Geographical Information Science (IJGIS). Tayor & Francis Publisher, November 2012. ISSN (Printed): 1365-8816 and ISSN (Online): 1362-3087.

[75] Ye, Z., Huang, X. J., He, B. and Lin, H. "Mining Multilingual Association Dictionary from Wikipedia for Cross-Language Information Retrieval" (28 pages), accepted by Journal of the American Society for Information Science and Technology (JASIST). Wiley InterScience

Publisher, March 2012. ISSN (Printed): 1532-2882 and ISSN (Online): 1532-2890.

[76] Li, C. and Huang, X. J. "Spam Filtering Using Semantic Similarity Approach and Adaptive BPNN", Neurocomputing Journal. Elsevier Publisher. 92:88-97, 2012. ISSN: 0925-2312.

[77] Chen, Y., Yin, X., Li, Z., Hu, T. and Huang, X. J. "A LDA-based Approach to Promoting Ranking Diversity for Genomics Information Retrieval'' (17 pages), to appear in BMC Genomics. June 2012. ISSN: 1471-2164.

[78] Zhao, J. and Huang, X. J. "Rewarding Term Location Information to Enhance Probabilistic Information Retrieval'', Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12), Portland, Oregon, August 12-16, 2012.

[79] Miao, J., Huang, X. J. and Ye, Z. "Proximity-based Rocchio's Model for Pseudo Relevance Feedback'', Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12), Portland, Oregon, August 12-16, 2012.

[80] Zhao, J., Huang, X. J. and He, B. "CRTER: Using Cross Terms to Enhance Probabilistic Information Retrieval'' (full paper), Proceedings of the 34th Annual International ACM SIGIR Conference on Research

and Development in Information Retrieval (SIGIR'11), Beijing, China, July 24-28, 2011.

[81] Zhou, X., Huang, X. J. and He, B. "Enhancing Ad-hoc Relevance Weighting Using Probability Density Estimation'' (full paper), Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11), Beijing, China, July 24-28, 2011.

[82] He, B., Huang, X. J. and Zhou, X. "Modeling Term Proximity for Probabilistic Information Retrieval Models'' (32 pages), accepted by Information Sciences Journal. Elsevier Publisher. March 2011. ISSN: 0020-0255.

[83] Ye, Z., Huang, X. J. and Lin, H. "A Bayesian Network Approach to Context Sensitive Query Expansion", Proceedings of the 26th ACM Symposium on Applied Computing, TaiChung, Taiwan, March 21-24, 2011

[84] Yin, X., Li, Z., Huang, X. J. and Hu, X. "Promoting Ranking Diversity for Genomics Search with A Relevance-Novelty Combined Model'' (16 pages), accepted by BMC Bioinformatics. March 2011. ISSN: 1471-2105.

[85] Xiangji Huang, Miao Wen, Aijun An and YanRui Huang. "A Platform for Okapi-Based Contextual Information Retrieval", Proceedings of ACM SIGIR 2006, Seattle, Washington, August 6-11, 2006.

[86] Miao Wen and Xiangji Huang. "A Multi-level Searching and Re-ranking Framework for Information Retrieval", Proceedings of 2006 IEEE International Conference on Granular Computing, Atlanta, USA, May 10-12, 2006.

[87] Xiangji Huang. YanRui Huang and Miao Wen. "A Dual Index Model for Contextual Information Retrieval", Proceedings of ACM SIGIR 2005, Salvador, Brazil, August 15-19, 2005.

[88] Xiangji Huang and YanRui Huang. "Using Contextual Information to Improve Retrieval Performance", Proceedings of 2005 IEEE International Conference on Granular Computing, Beijing, China, July 25-27, 2005. ISBN: 0-7803-9017-2 and IEEE Catalog Number: 05EX1036.

[89] Xiangji Huang. "Incorporating Contextual Retrieval into Okapi", Proceedings of ACM SIGIR Workshop on IR in Context (IRiX'05), Salvador, Brazil, August 19, 2005. ISBN: 87-7415-290-4

Appendix

Appendix I. Code for *Figure 2.*

The shapes of the three influence functions on relative term frequency:

```
plot(x=NULL,y=NULL,xlim=c(1,10), ylim=c(0,1.5), xlab="Term Frequency",
ylab="Influence")
x=c(0,2)
y=c(0,0)
lines(x,y)
x=c(2,8)
y=c(0,1)
lines(x,y)
x=c(8,10)
y=c(1,1)
lines(x,y)
curve(((x-2)^3)/(6^3),2,8,add = TRUE, col = "blue")
curve(((x-2)^1.5)/(6^1.5),2,8,add = TRUE, col = "green")
leg <- c("liner","quadratic","cube")
    legend('topright', legend=leg,lty=1, col=c('black', 'green', 'blue'), bty='n', cex=.75)
```

Appendix II. Code for *Figure 15.*

Performance Comparison on MAP

```
filename1 <-
"C:\\Users\\fengtao\\Desktop\\Yorklab\\Experiment\\BM25AffectedByAverageTF\\d
ata\\Blogs06\\1Linear.P10.scan"
filename2 <-
"C:\\Users\\fengtao\\Desktop\\Yorklab\\Experiment\\BM25AffectedByAverageTF\\d
ata\\Blogs06\\2Quadratic.P10.scan"
filename3 <-
"C:\\Users\\fengtao\\Desktop\\Yorklab\\Experiment\\BM25AffectedByAverageTF\\d
ata\\Blogs06\\3Cube.P10.scan"

c1 <- read.table(filename1, header=FALSE)
c2 <- read.table(filename2, header=FALSE)
c3 <- read.table(filename3, header=FALSE)
x <- seq(0, 20,by = 1)
all <- c(c1[,1],c2[,1],c3[,1])
r <- range(all)
plot(x=NULL,y=NULL,xlim=c(1,20), ylim=c(r[1],r[2]), xlab="Blogs06",
ylab="P@10")
leg <- c("linear","quadratic","cube")
legend('topright', legend=leg,lty=c(1,2,4), pch=c(0,1,2),col=c('black',
'green', 'blue'), bty='n', cex=1.20)
lines(x,c1[,1],type="o",pch=0,lty=1,col="black")
lines(x,c2[,1],type="o",pch=1,lty=2,col="green")
lines(x,c3[,1],type="o",pch=2,lty=4,col="blue")
```

Appendix III. Code for *Figure 16.*

Performance Comparison on P@10

```
filename1 <-
"C:\\Users\\fengtao\\Desktop\\Yorklab\\Experiment\\BM25AffectedByAverageTF\\d
ata\\WT2G\\1Linear.P10.scan"
filename2 <-
"C:\\Users\\fengtao\\Desktop\\Yorklab\\Experiment\\BM25AffectedByAverageTF\\d
ata\\WT2G\\2Quadratic.P10.scan"
filename3 <-
"C:\\Users\\fengtao\\Desktop\\Yorklab\\Experiment\\BM25AffectedByAverageTF\\d
ata\\WT2G\\3Cube.P10.scan"

c1 <- read.table(filename1, header=FALSE)
c2 <- read.table(filename2, header=FALSE)
c3 <- read.table(filename3, header=FALSE)
x <- seq(0, 20,by = 1)
all <- c(c1[,1],c2[,1],c3[,1])
r <- range(all)
plot(x=NULL,y=NULL,xlim=c(1,20), ylim=c(r[1],r[2]), xlab="WT2G", ylab="P@10")
leg <- c("linear","quadratic","cube")
legend('topright', legend=leg,lty=c(1,2,4), pch=c(0,1,2),col=c('black',
'green', 'blue'), bty='n', cex=.75)
lines(x,c1[,1],type="o",pch=0,lty=1,col="black")
lines(x,c2[,1],type="o",pch=1,lty=2,col="green")
lines(x,c3[,1],type="o",pch=2,lty=4,col="blue")
```