SDPR: RESEARCH PAPER RANKING USING SEMANTIC RELATEDNESS

By

CHINMAY MURUGKAR

(Under the Direction of Lakshmish Ramaswamy)

ABSTRACT

Researchers all over the world everyday have to search through large amounts of research papers and filter them to choose the right candidate. Ranking these documents is the easy way to mitigate the efforts and time consumed to filter out the unwanted papers. Lexical cohesion is the property of the text that we use to find coherence in the document. This coherence, in turn, can be used to find the semantic relationship between the query words and the documents. We present SDPR: an efficient search framework for ranking research papers. Three significant features characterize this system. First, classifying the words according to their importance in the query. Second, we weight the semantic distance between query and the document using spread activation technique in wordnet graph based ontology. Third, in order to ensure the quality of the rank, we weight each document based on the term frequency of query words in them. Results demonstrate a considerable amount of improvement over traditional keyword-based searching algorithm and helps in query disambiguation.

INDEX WORDS: Semantic Distance, document ranking, wordnet, Node spread activation

SDPR: SEMANTIC RELATEDNESS FOR DOCUMENT RANKING

by

CHINMAY MURUGKAR

B.E., Nagpur University, India, 2009

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2013

SDPR: SEMANTIC RELATEDNESS FOR DOCUMENT RANKING

by

CHINMAY MURUGKAR

Major Professor:	Lakshmish Ramaswamy
Committee:	Hamid Arabnia
	Ismailcem Budak Arpinar

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2013

DEDICATION

To my parents, family & friends, for their love, support and encouragement.

ACKNOWLEDGEMENTS

Past 3 years are the most enriching years of my life and has been a big learning curve. I take this opportunity to thank Dr. Lakshmish Ramaswamy for his constant support, feedback, encouragement and motivation. I would like to thank Dr. Budak Arpinar for his constant inputs, guidance and motivation for my project. I would also like to thank Dr. Hamid Arabnia for providing his valuable inputs and motivation for my project.

I would also like to thank my colleagues Akshay, Aniruddha, Rohit & Muthu for all the support and motivation. A special thanks to Dr. Ravi Prasad for their love, support, motivation and giving me an opportunity to work as lead web developer for the UGA Center for Applied Isotope Studies.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In recent 10 years 9.4 million articles, notes, and reviews, published in roughly 11,000 indexed journals and the citations received through these journals are roughly 85 million [1]. As Research papers published every year are increasing exponentially, it is challenging for researchers to filter through huge number of papers to get to the right candidate, which can be time consuming and tedious. Also, the popularity of the research paper based search engines such as CiteSeer and IEEE Xplore [80] is increasing, not just for searching literature but also for making hiring decisions. Thus, accuracy of these systems becomes very significant.

Searching on research literature platform can be very different than searching document on World Wide Web. It poses a very unique challenge that demands for semantic and syntactical analysis of the papers to understand the underlying context of the document while ranking. Various models were studied previously[66][70][77] that considered citations in the paper as the base to rank the documents including paper rank and page rank algorithms. But these models fail to take the factors like semantic and syntactic analysis into consideration that highly vary from user to user according to their opinion. For example, popular research paper search system like Google scholar tries to resolve this problem by using 'citations' in the paper as parameter[66] to rank them. The weight of the paper is decided on the number of times the paper has been cited in other papers and the relative weight of the papers that are cited in that document. Some of

the groundbreaking research papers that date back when the literature knowledge base was not as large as today can lack highly weighted citations or have minimum citations in the paper. Also, citations in the paper vary according to the field of the paper, for example, papers related to philosophy consists of more number of citations than compared to the research papers published under mathematics domain [70]. Thus, we believe that it is very important to consider lexical analysis while ranking the papers.

Factors such as relevance, cognition, interaction and propinquity in the words [76] play a significant role in deciding the right candidate for the user. Thus, while creating a powerful searching tool for research papers, it is very significant to consider following features:

- Research papers typically consist of 3,000 to 10,000 words [81] and thus lexical analysis is very significant in order to understand the coherence between the words in research paper.
- While performing lexical analysis, endorsement of the system to search. This demands for establishing the measure to compute the semantic relationship between the words in the query and the documents.
- Synonymy - Any concept, which is appearing in the document, can be similar to many of the other concepts in the document. For example: The word 'cancer' would appear in the documents in different formats
- Polysemy - The words that are appearing in the document can be used in many different ways.

Many systems that work on numerical statistic method have tried to work around the problem of polysemy using the term weighting [10], matching the query to the text in the document using the keyword relations [6] or Latent semantic Indexing [22][8], but these system don`t consider synonymy in the sentences. Some systems try to use framework for tag-based research paper[78][82], where tags in paper are used to determine relevancy.

The approach of term weighting which is also know as 'term frequency' i.e. TF proposed by Luhn [10] was complemented by Spärck Jones`s [2] introduction of Inverse document frequency which took into consideration, how many documents are actually related to a particular term. The idea of combining these two ideas [5][7] lead to tf-idf combined matrix [8] which was developed using singular value decomposition (SVD). The SVD of large sparse matrices of term by documents was created to associate them with each other and rank the documents. This method is also known as Latent Semantic Indexing (LSI) as it shows important relationship between term and document, which cannot be recognized individually. Thus, term weight can have a huge impact over the ranking of the document.

This thesis presents SDPR, an efficient search framework for ranking research papers. We are dividing our system in three steps viz., classifications of the query words, semantic distance calculator and weight calculator. First of all in effort to understand the significant words in the query, we classify query words based on their frequency in all of the documents. Secondly, to find the relation between user query and the document we have to understand the relationship between words. This demands for

understanding the context of the document and analyze the semantic relationship between the query words and document words. Thirdly, we present weight calculator to deicide the weight of the document based on term frequency of query words in the document. Weight calculator would endorse the ranking of the document in addition to the semantic weight.

We have performed a number of experiments to measure the efficiency in document ranking. Results demonstrate a considerable amount of improvement over traditional keyword based searching algorithm and helps in query disambiguation.

In the following chapters, we would peek into this model and its background in some more detail.

CHAPTER 2

Motivation and Contribution

2.1 Motivation

The citations based, tag based and graph based systems[77][78][66][83][67][69] are the motivating factor behind our work in this thesis. Here we would highlight the factors that set apart the models that are based on semantic analysis from that of citations and tag-based algorithms. The underlying algorithm that contributes in searching for a popular research-paper searching tool, Google scholar, is PageRank algorithm[83]. PageRank considers the fact that each of the documents is related to each other on World Wide Web through the links in it. Thus, a Page Rank [hereby referred as PR] depends on probability that the user would access a particular link through that page, which is referred to as 'page hit'. In case of Google scholar PageRank tries to do this using citations in the paper. Rank of a paper is decided based on number of times a paper is cited and weight of the papers cited in that paper. Even though citation being a very common form of weighing for the rank of papers, we believe that in many cases it can be misleading. Citations can pose a problem due to different factors, which are as follows:

- Many groundbreaking research conducted during the period where the literature knowledge base was not broad can have few citations in them. This can lead to decrease in PR of the paper

- Context decision[83] can be a problem while deciding rank based on the citations, as rank assigned to a particular citation can be based on ranks of paper in which it is cited. Transitively, rank assigned to a paper can be a projection of citations in that paper and the rank of the papers in which they are cited.
- Number of citations in the paper differs according to the field of the research paper[81]. Some fields such as philosophy can have relatively more citations than it does in Mathematics research paper.
- Multiple copies of research papers is one of the problem that may arise due to citation based ranking[68]. As many authors deny indexing their paper, it can be hard to reach the paper without performing a text content analysis.

Considering all these factors we believe that taking into consideration, citations is not sufficient for efficiently ranking the documents. It is significant to understand the context of the paper. The system that we propose tries to overcome these challenges by examining the semantic relationship between the words in the query and that of the documents, which help to weight and rank documents accordingly

2.2 Contribution

During lexical analysis of the paper, it is significant to analyze the context of the words in the document [20][47]. Our proposed approach helps to understand context by analyzing the relationship between the query words and the words in the paper and ranking them accordingly. We have the following significant contributions:

- Query words classification

  Importance of the words appearing in the query depends on the frequency of the query words in the all of the documents[2][5]. If the frequency of a query word is high in all of the documents, it can lead to close rank score of the documents. Thus, we have designed a word classifier to separate out important words from non-important words in the query.

- Semantic distance calculations

  In order to find the semantic similarity between query and documents, it is significant to understand relationship between the words in them. We use wordnet api to derive the semantic distances.

- Term weighting

  The term frequency can play an important role in deciding the rank of the paper[6][7][8]. Thus, we believe that ranking the documents w.r.t term frequency can empower the existing semantic scoring system to provide better results. To decide this we have a weight calculator that we would be considering in chapter 6. Documents weight would be decided based on the frequency of the words in that document and its weight among all the words in the query.

2.3 Organization

In this chapter 3 we would take a look into the building blocks of SDPR. We would look into each component in different sections of this chapter and try to realize the significance and its function in detail.

Chapter 4 explains the process for classification of the words in the query, which is required to differentiate between significant words from that of non-important words. Also, we will see pre-processing methods used for filtering and refining the query.

In order to rank the documents we are finding the similarity between the query and the research paper using the relationship weights between them. To do this we would find the coherence between the words. This is explained in more detail in Chapter 5.

Chapter 6 explains the lexical analysis, which helps in endorsing the semantic weight of the document in order to get better ranking.

In chapter 7 we would peak into the experimental setup we used in order to test the results. We also plot graphs by considering different parameters such as Precision, Recall, False positive, False negative and F-Measure.

In Chapter 8 and 9, we would discuss related work and the work that we would be doing in future.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 Introduction

In this chapter we would take a look into the building blocks of SDPR. We would look into each component in different sections of this chapter and try to realize the significance and its function in detail. The architecture of SDPR contains of three major blocks. First, Document preprocessing which is responsible for filtering the documents, categorizing them, storing them in database and finally indexing all of them at document level. Second component is query pre-processing which helps in filtering the query and elaborating the query using its synsets. Third, Research Paper ranking which is significant block of the architecture that helps in ranking documents by weighting words with semantic relationships and the term weight. Each of this block is explained in detail as follows.

3.2 System Architecture

Figure 3.1 represents the System level architecture of SDPR. It is explained as follows.

3. 2.1 Document Preprocessing

Document preprocessor is responsible for storing and indexing documents. Document normalizer gets the document and tries to filter out all the non-ascii characters, which are not required for indexing and searching. After converting the document into a plain text string it is stored in a data repository i.e. DBLP Ontology. Documents are stored

with its title, abstract and body classified in the system. All of these documents are sent to Document Indexer where they are inverted indexed at document level and stored as a flat text file on the disk.

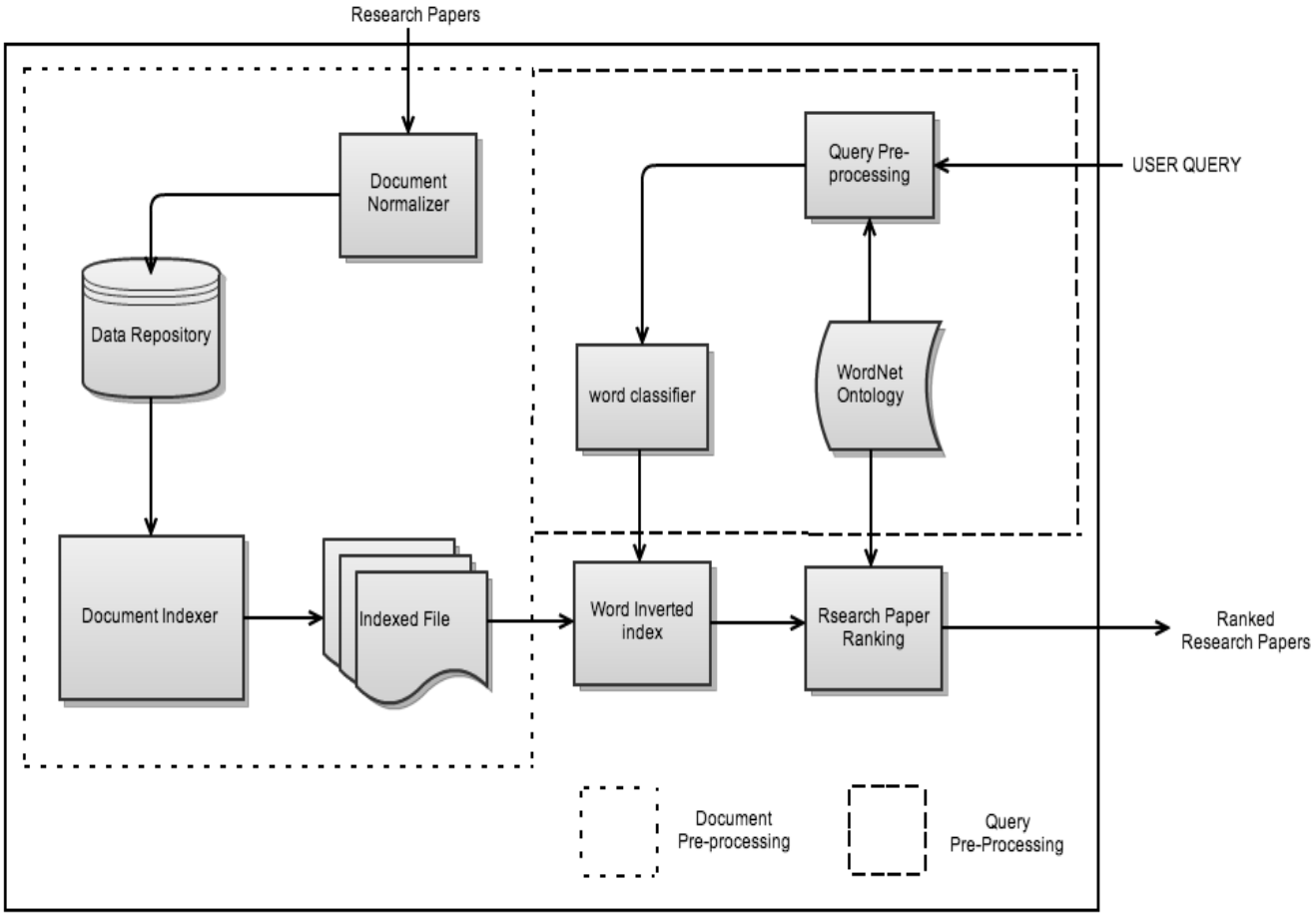

FIGURE 3.1: system architecture

3.2.2 Query Pre-processing

Query preprocessing is one of the important components in the system that plays crucial role in word classification algorithm. This component is responsible for forming systems and classifying query. While pre-processing the query we will filter out all the unnecessary prepositions, special symbols or non-ascii characters which are not helpful for ranking. To establish polysemy in the algorithm, we take out each word from the query to form a synset. Synset is a special set that contains all the synonyms related to a particular word. We derive these synonyms through wordnet graph ontology using Rita wordnet API. This set would help solving the problem of polysemy while performing semantic processing on document. We would discuss this in Chapter 4 in more detail.

Classification of words

Word classifier is responsible for grouping the words, which are important for searching and the one that are less important than others. Weighting each of the word according to its significance in the query is one of the important factors to identify the lexical cohesion and for this purpose we would be using term weighting method based on the frequency of these query words in research papers.

3.2.3 Research paper Ranking

This is the third and crucial component of the system. SDPR consists mainly of two different components i.e. semantic distance calculator and weight calculator. Lexical

cohesion is the property of the text using which we can find the coherence in the sentences in the document with the help of deictic elements in them.

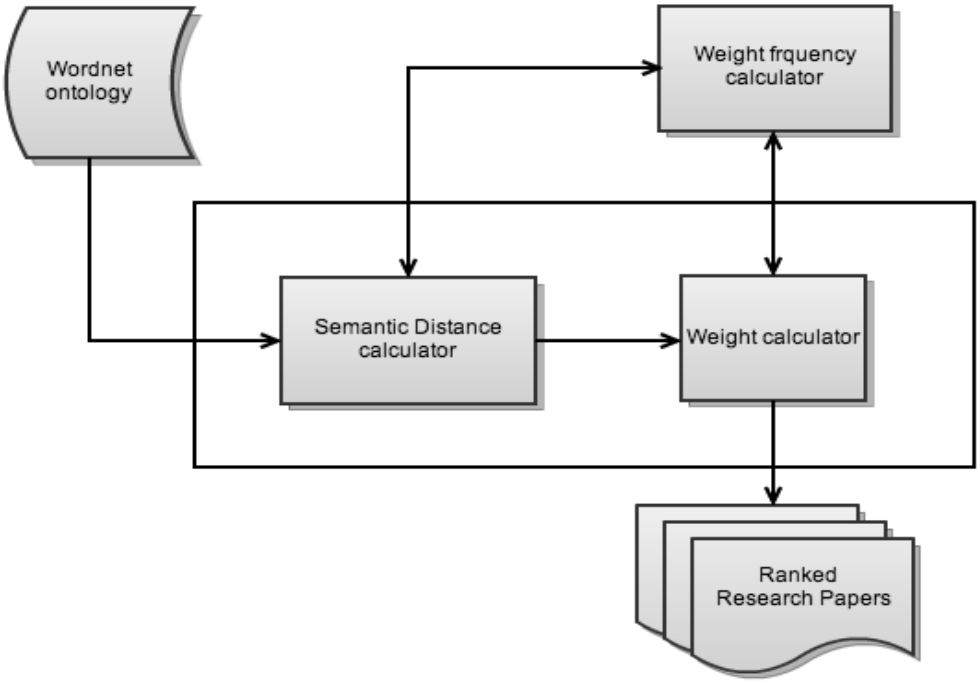

FIGURE 3.2: Document ranking system architecture

Deixis is the phenomenon where to understand the meaning of certain words or phrase requires the understanding of the context of the sentence. We achieve this through semantic distance calculator by weighting the semantic relationships between the words in the query and research paper. The relationship weight is decided based on the node hop distance between the two words in the wordnet graph based ontology where every word is related to other word based on its context and part of speech. Thus, we

resolve the polysemy and synonymy with help of this model. We would look into semantic distance calculator in more detail in chapter 5.

To ensure the quality of the ranking results we have added term weighting to the model. We try to weight each document based upon the frequency of the query words appearing in the document. The semantic weight and term weight both add up to form the weight for research paper. This weight is then stored in a hashmap and then we sort it to get ranked research papers. We would dive into understanding of weight calculator in more detail in chapter 6.

CHAPTER 4

WORD CLASSIFICATION AND PRE PROCESSING

In this chapter we will look at the significant components of the system responsible for filtering and identifying important words in the query. Before moving towards word classification we would brief the preprocessing that is undertaken on the query.

4.1 Pre-Processing

We took into consideration various pre-processing methods to apply on query such as word boundaries, lemmatization, stop word removal, case folding, word token, Synset formation and Indexing. Let us look into each of these what do they mean and how do we deal with them.

Word boundaries: In this we try to remove the white spaces and punctuations but the few words like isn`t, I'll or I've are really hard to deal with so we deal with few of them using dictionary of these words that we need to remove or keep them as it is.

Lemmatization: This is also known as stemming. Many times words are not really required if specially We find out the root word of the query word and add it to original query. We use wordnet API to find out the stem of each of the word.

Stop word removal: Most frequent words do not carry much meaning such as a, of, the and so on. We create a dictionary of all the stop words that we need to remove from the query words and while processing the query we remove the words from it that are appearing in the dictionary.

Case folding: Generally while processing the documents and query we would not be need the words in the case they are appearing. Also, this can pose a problem while searching through the documents and while indexing them. The two similar words but with different cases can be indexed differently if we treat it case sensitively. So, for this system we prefer to lowercase all of the characters in the query as well as the documents.

Synset formation: This is one of the important factors in the system. Synset formation is nothing but finding out the array of words which synonyms for a particular word. This would help in treating the documents containing words of the similar meaning equivalently. We find all the synonyms using RITA wordnet API.

For Example: User query might contain a word 'cancer' but while searching for the documents related to cancer it is also necessary to observe the documents that contain words which are equivalent or closely related to cancer like tumor or malignant growth. The Synset would look like this:

Cancer <tumor, malignant, carcinoma, canker, blight>

We would add this synset to the original query to form a new query.

In each of the above preprocessing step we keep on adding the resulting outcome to the original query to get the final query word set 'S_m' which we would be using in our further calculations for weighting and ranking the documents.

4.2 Word Classification

Now let us move towards understanding word classification. This process would help to separate out the important query words that help in term weighting process. A word

that appears in most of the documents can prove to be less important for weighing the research paper[12][2][5]. For example: a word 'data' can occur in most of the documents[85] so if we assign score to the paper based on the weight calculated using this word it would be deceptive or equivalent and would not help much in ranking documents. So, our first problem is to decide which words are really important out of these query words and to solve this problem we use the frequency of these words and how many times its appearing in the documents. To classify the words we use term frequency as base to understand importance of a word in overall documents. We index the words in query and store it as a hash map with word as key and frequency of the word over all papers as value. We then compute the mean of frequencies of all query words stored in the hash map to decide the Frequency threshold 'F_T' for decision. F_T is used as a decision parameter for classification. We divide the words in two sets i.e. set 'S_Q' that have frequency lower than the F_T and set 'S_L' that have frequency higher than the F_T.

Thus we define the two sets as follows:

$$\boldsymbol{S_Q} = \{\boldsymbol{a} \mid \boldsymbol{f_a} < \boldsymbol{F_T}, \boldsymbol{F_T} > \boldsymbol{0}, \boldsymbol{F_T}, \boldsymbol{f_a} \in \boldsymbol{N_1}\} \quad \text{-(4.1)}$$

$$\boldsymbol{S_L} = \{\boldsymbol{a} \mid \boldsymbol{f_a} > \boldsymbol{F_T}, \boldsymbol{F_T} > \boldsymbol{0}, \boldsymbol{F_T}, \boldsymbol{f_a} \in \boldsymbol{N_1}\} \quad \text{-(4.2)}$$

Here, 'a' is element that belong to the set and 'f_a' is the frequency of element 'a'

Words that have lower frequency than threshold would prove to be important and thus words in set S_Q are significant for document weight computations.

Frequency threshold 'F_T' for classification can be defined as follows:

$$\boldsymbol{F_T = \frac{\sum f(w_q)}{n_q}} \qquad -(4.3)$$

Here, 'w_q' is a word from query set 'S_m' and 'n_q' is number of words in the set

For example if we have a query and apply these all processes on it would look something like this:

Query: Use of malware analysis for improving data security.

After applying: word boundaries, lemmatization, stop word removal, case folding, word token, Synset

<use, application, employment, malware, malicious logic, malicious software, anatomizing, breakdown, deconstruction, data, information, secure, protection, defense>

Word classification process:

Table 1: Word classification process

Word	**Frequecy**
malware	45
analysis	110
data	150
security	86
use	130

Now, considering the data above and applying it to the equation for mean, we get:

$mean = |\frac{\Sigma f(w_q)}{w_q}| = 104$

Thus, we get two sets i.e.

Important words : <malware, security>

unimportant words : <analysis, data, use>

Thus this effectively helps in separating out the words in the query. In following chapters we would peek into the significance of the set S_Q in weighting the terms in the documents and compute the weight of each of the document for ranking.

CHAPTER 5

SEMANTIC DISTANCE CALCULATOR

5.1 Introduction

In this chapter, we take a brief look into the process of measuring the semantic relationship between query and the document. Cohesion is responsible for giving the sentence or text meaning by establishing grammatical or lexical links between the words[86]. To find the semantic relationships between the words in the query and the words in the paper, it is significant to find the coherence in the words. To achieve this we are interested in lexical cohesion, as it would help us compute the semantic relationships between the words. First, we analyze the context of the document. Second, we will compute the semantic weight for relationship between the words. Let us look into the Semantic distance calculator algorithm.

5.2 System Algorithm

Presented in figure 3 is the Semantic distance calculator algorithm. It is very important to find out the relationship between the words for lexical analysis. This can determine the semantic relatedness between the research paper and the query. We start with analysis of the context of the paper. We make an assumption that most frequent words appearing in the document would define the context. After trial and error method we concluded that the first ten most frequent words could be the best representative of the context of the paper. Let the set of these words be denoted by 'S_T'. To do compute the semantic closeness of the query and the document we consider each of the word in the

set S_T and its semantic distance with the words in the set S_B. To find the semantic distance between the words we use wordnet activation spreading technique. Now let us take a brief look into wordnet activation spread technique and the process to access it using Rita Wordnet API:

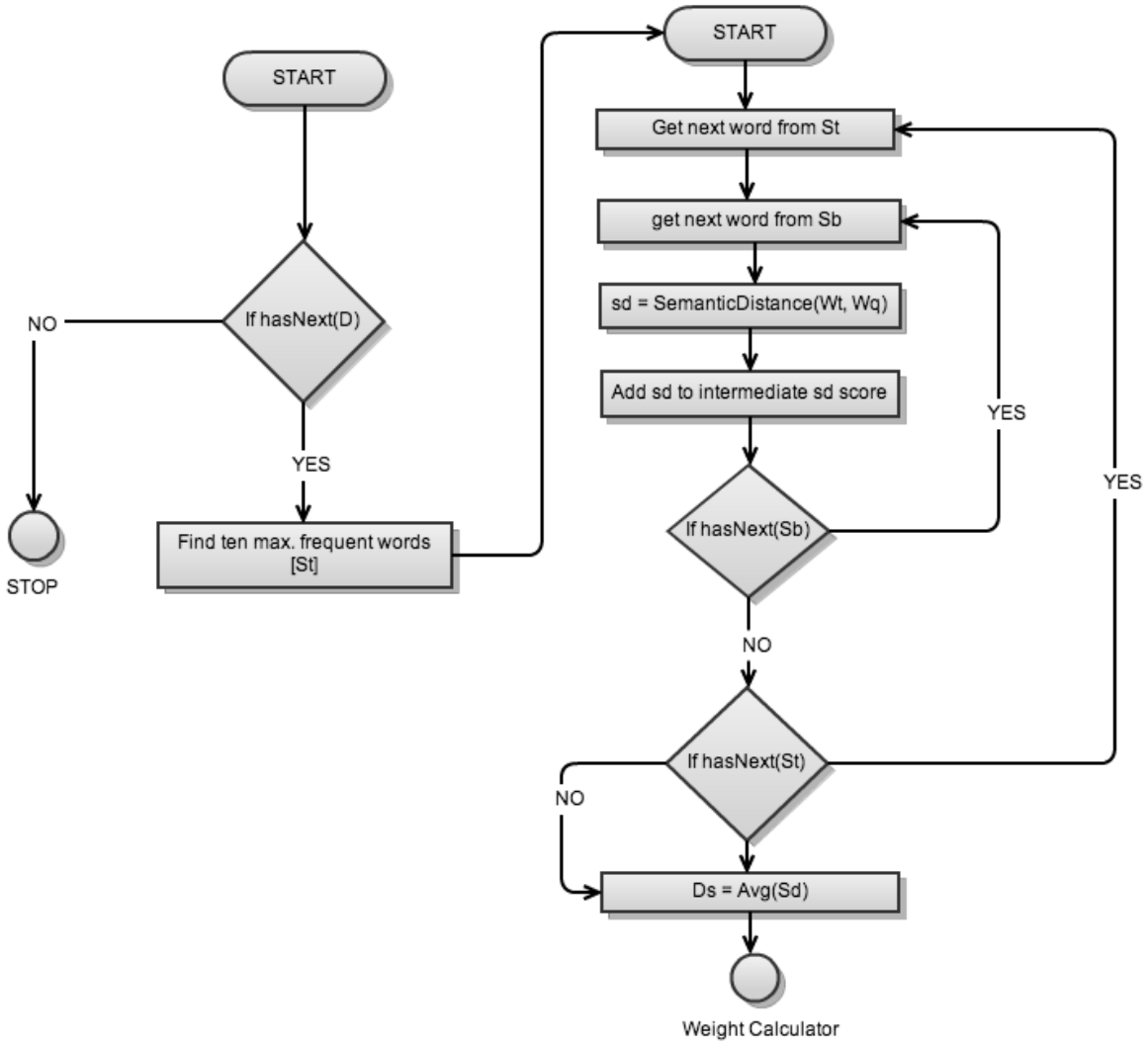

FIGURE 3: Semantic distance calculator algorithm

Spread Activation technique [64] aims at finding the coherence between words using the deictic elements. Two words, w_x and w_y, for which we have to compute sematic distance are located on the graph. Common parent is then located for both words and distance is calculated by measuring the nodes required to hop from word w_x to word w_y through parent word w_p. This helps in reducing the lexical dependence on the terms in the Q&A's, which allows a degree of lexical variation. Our semantic distance calculation is based on WordNet [65], a semantic network of English words and phrases built at Princeton University. Wordnet is a graph of words connecting each other according to the relations exists between them. While relating words it takes into consideration the synset of the word where synset contains words interchangeable in context. Wordnet is divided in four subnets according to their part of speech i.e noun, verb, adjective, and adverb.

We use RitaWordnet API to access wordnet graph and find semantic distance. RidaWordnet API takes word and part of speech (pos) of the word as parameter. The 'pos' function gives back the part of speech encoding of the particular word in the wordnet, which is in turn used to find out the senses of the word in the graph using polysemy function. Now, the nodes are activated unless we reach the common parent for both the words and then find out the semantic weight. For instance the word 'play' can be used in different context as a play in theater or play in music or play in sports. It tries to find out the sense of the word first and then activates the respective nodes. The scoring system varies between 0 through 1 where 1 being the score that represents no

relation and 0 the strongest relation. Thus, for example words like play and fridge are not related and thus, return score as 1 and the words play and football return 0.4

Thus, this is the process through which semantic distance is calculated. We compute the semantic distance between the query term 'w_b' and the words in the set S_T in the document. This would determine the semantic closeness between the query and document. We consider each of the word in the S_B and find its semantic distance with the all the words in S_T. Then, we take average of this distance and store as an intermediate semantic calculation. We repeat the same process with rest of the terms in the document. This can be computed as follows:

$$\boldsymbol{I_s} = \boldsymbol{sd}(\boldsymbol{w_t}, \boldsymbol{w_b}) \;\; where, w_t \in S_T \, , w_b \in S_B$$

Here, I_s represents semantic distance between a single term w_t of word set TenMax i.e. S_T and the single term w_b of query set terms i.e. S_B.

$$\boldsymbol{I_d} = \frac{\sum_{i=0}^{n_b} \boldsymbol{I_{s_i}}}{\boldsymbol{n_b}} \, , Where \; n_b \; is \; the \; number \; of \; terms \; in \; Query \; Set$$

In the above equation I_d represents the intermediate semantic distance, which is the mean of the semantic distance between a term in the S_T term set and all the terms in query set. We then compute the Semantic distance between the query and document with following equation:

$$\boldsymbol{D_s} = \frac{\sum_{i=0}^{n_t} \boldsymbol{I_{d_i}}}{\boldsymbol{n_t}} \, , Where \; n_t \; is \; the \; number \; of \; terms \; in \; S_T \; Set \qquad -(\mathbf{5.3})$$

In the above equation D_s is the final semantic distance between the query and that document. This semantic distance represents the mean of the intermediate semantic distances that is mean of all the intermediate semantic distances I_{d_i}.

We store the score D_s as intermediate weight, w_i, for paper.

CHAPTER 6

WEIGHT CALCULATOR

6.1 Introduction

To ensure the quality of the rank of the research paper we consider weighting each paper by considering the frequency of the terms in set S_Q in it. In this chapter we take brief look into the algorithm of the weighting each research paper. Considering document weight based on term weight would help to find out the similarity between the research paper and query. This implies that if the paper would be in interest of the user if the term in the query were to appear in it. Combining the semantic distance with the term weight would provide us with a strong document weight to rank them within corpus.

6.2 System Algorithm

Along with the semantic relationships we are considering term weight to make the relevance decision stronger. Here we are interested in the numerical statistic of term frequency for weighting documents as this would help in performing lexical analysis on the document. Presented in the figure 4 is the algorithm of weight calculator. For each of the document D_i we create an index of the query terms and the terms in the set S_T i.e. set of top ten most frequent words in the document. This would help to recognize the context of the document. We would now take a brief look into the computations for weighting each document D_i.

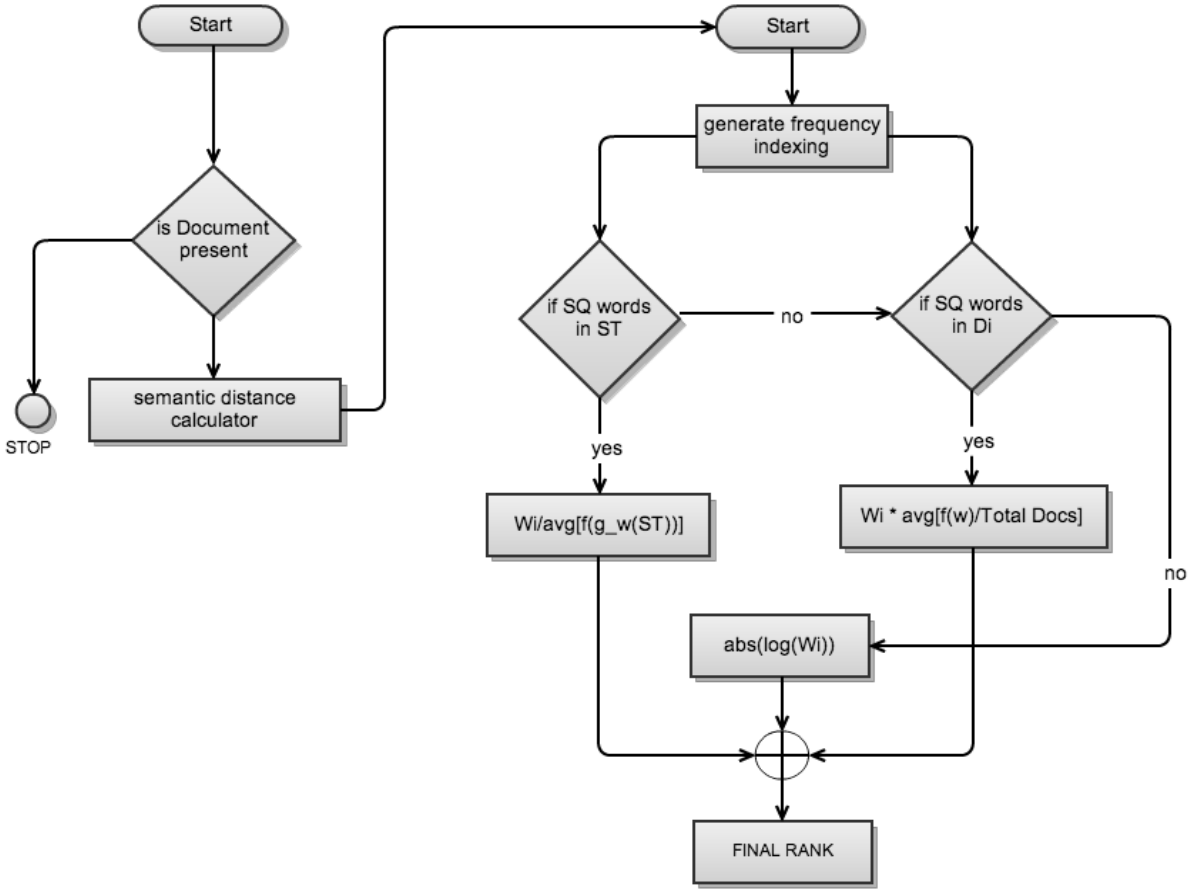

FIGURE 4: Semantic distance calculator algorithm

Weights are decided based on the words appearing in the document and that compared with the terms in the set S_Q that we have already created during the query word classification step.

The words in set S_T derived from a research paper would prove to be very important while deciding context[5]. Also, in the step of classification of words, we have

recognized two sets of words from the query terms, the one that are important words S_Q and the others that are non-important words S_L.

Thus, while considering term weight, we try to analyze the frequency of selected terms from the query to emphasize the importance of the document by calculating respective weight. We take exhaustivity and specificity into consideration while designing the functions to weight D_i. Exhaustivity is number of terms a document contains and specificity is the number of documents in which the term contains. Both of these parameters would help in determining the importance of the query terms in the document.

Let us consider that S_a be the set of words present in set S_Q and are appearing the document D_i. If any words in the S_Q set were occurring in the S_T set, then we calculate the weight $\boldsymbol{w_d}$ for that document as:

$$\boldsymbol{f_s} = \frac{\sum_{i=0}^{n_a} \boldsymbol{f}(\boldsymbol{w_{a_i}})}{\boldsymbol{n_a}} \qquad -(6.1)$$

Where, $\boldsymbol{f_s}$ is the average frequency of the words. $\boldsymbol{w_a}$ is the word from set S_a and n_a is the number of terms S_a contains.

$$\boldsymbol{w_d} = \frac{\boldsymbol{w_i}}{\boldsymbol{f_s}} \qquad -(6.2)$$

Here, '$\boldsymbol{w_i}$' is the intermediate weight that we have stored in the step of semantic distance calculation for that document.

If the above condition of having S_Q set words occurring in S_T is not satisfied then we check if S_Q words are appearing in rest of the document. We have assumed that the words appearing in S_T set represent the context of the document. Thus, the words other

than S_T, which are in the document and also appearing in the S_Q set of words, are very important. But we also have to decide the weight for the words, which appear in rest of the document and not in S_T set. Thus, we consider the weight of the words with respect to all the documents to understand its importance in D_i.

Let n_d be the total number of research papers in data repository. Let S_b be the set of words appearing in D_i. Thus, to calculate the weight of the document we equate it as follows:

$$\boldsymbol{f_d} = \frac{\sum_{i=0}^{n_b} \boldsymbol{f}(\boldsymbol{w_{b_i}})}{\boldsymbol{n_b}} \qquad -(6.3)$$

Where, $\boldsymbol{f_d}$ is the average frequency, $\mathbf{w_b}$ is the word from **set** $\mathbf{S_b}$ and $\mathbf{n_b}$ is total number of documents in data repository

$$\boldsymbol{w_d} = \boldsymbol{log}\ (\boldsymbol{w_i} * \ \boldsymbol{f_d}) \qquad -(6.4)$$

In the two cases above the average frequency of the words signify the lexical context of the document. As in our equations we are taking the ratio of Intermediate score i.e. semantic distance to the average frequency of the words appearing in the documents, the weight would represent the similarity between the query and the document. As, the semantic distance would decrease and frequency would increase the overall weight that the document would gain is decreased. In our model the decreased document weight signifies that it is more important. Thus, this suggests that as per the occurrence of the query words in different condition would affect the contextual analysis of the document affecting its weight. Also, if the semantic distance is increased between the words and the frequency is decreased, the weight is increased, thus reducing the rank of the document.

If both of the above conditions fail we compute the weight $\boldsymbol{w_d}$ as follows:

$$\boldsymbol{w_d = 1 - abs\big(log(w_i)\big)} \qquad -(6.5)$$

Thus, we conclude with the computation of Final rank 'r'. Which is defined as follows:

$$\boldsymbol{r_i = w_d} \qquad -(6.6)$$

The final rank is added to the hash map (η) where key is the index i of the document D_i and value stored for corresponding key is the rank r. Thus it can be represented as follows:

$$\boldsymbol{\eta \rightarrow (i, r_i)} \qquad -(6.7)$$

We sort the values in the hash map i.e. $\boldsymbol{r_i}$ in ascending order to give the final rank of the research papers.

CHAPTER 7

RESULT

In this section we study the experimental evaluation and experimental setup for testing the system. Due to various ways for the interpretation of the results of research paper ranking system, we evaluated it using nine human subjects. Where all the nine subjects were graduate students of computer science department and not aware of the familiar with the research presented in this thesis. Human subjects were provided with the document that contains title and the abstract of one hundred and thirty three research papers and a query. They were asked to identify the relevant papers to the query in the document provided to them. These results were then compared with the result set from our system to compute the quality.

7.1 Experimental Setup

7.1.1 Model Used

To test the functionality of this model, we tested SDPR model against popular key word based searching model TF-IDF [Term Frequency – Inverse Document Frequency]. Many researchers [2][5][74] have studied and proved the competency of the results for this system in past. It is also used in the Apache Lucene Solr, which is used commonly as a search API in web application and system applications.

7.1.2 Query set used

For our experimental purpose we have identified 9 queries out of which we have divided queries in different categories based on following factors:

- Use of words which are common in most of the documents
- Use of acronyms in the query
- Use of words which have less or none coherence with rest of the words in query
- Use of the words with no synonyms related to it, mostly these are the words that are noun eg : Hadoop.

7.1.3 Dataset Used

We are using research documents, which are related to computer science field. All the documents are stored in DBLP ontology. We downloaded DBLP bibliography [75] from its original XML format to RDF/OWL format. Some of the properties that were used to store the document for experimentation are:

Classes

- Citation
- Book
    - Extends: Document
    - Datatype Properties: datatypeField
- Publisher

Datatype Properties

- booktitle

Object Properties

- url

Thus, we store these document and index all the documents in a flat text file using inverted indexing technique. The statistics of the dataset are as follows:

Table 2: statistics of the dataset

Number of documents stored	Size of files stored on disk	Number of queries
133	7.1 Megabytes	9

7.1.4 RELATED APIs and Ontologies

WordNet Ontology:

We are using WordNet ontology for finding the semantic distance. WordNet ontology is in RDF format. Ontology is in XML graph format, which is then converted to RDF type.

WordNet Ontology Storage space: 15 Kilobytes.

RITA WordNet API:

Rita WordNet API is used to access the WordNet graph. The specifications of the API used for this experimentation are:

Table 3: Rita Wordnet API setup

Version	Alpha[032]
Tested	Processing 1.x
Jar file used	Core1.0.jar

7.1.5 System Specifications

To test SDPR against the TF-IDF algorithm and for running all related APIs we have used machine with following specifications:

Table 4: Machine specifications

Processor	2.3 GHz Intel Core i5
Memory	4 GB 1333 MHz DDR3
Operating system	Mac OS X Lion 10.7.5

Following are the Java version specifications:

Table 5: Java version specifications

Java version	1.0.6_37
Java HotSpot(TM)	64 bit server VM
Max Heap size	123.938 Megabytes

We have collected the data from nine different users related to each of the query to test the document ranking correctness. Data provided to users is: Test query and document containing all the research papers with information as Title followed by related abstract. Users were asked to mark the related documents to the query by reading though the document provided. Then, these results were considered to verify against two documents for the same query i.e. document containing results obtained from TF-IDF algorithm and document containing the results obtained from SDPR algorithm. To

compute the result performance we considered following parameters for each of the query:

- Precision
- Recall
- False positive value
- False Negative value
- F-measure or F- precision or F – score

7.2 Precision And Recall

Precision and recall are widely used in IR systems to test the results. We have computed and plotted the test results for each of them.

Precision and Recall are defined as:

Precision = number of relevant documents / number of retrieved documents

Recall = number of relevant documents in result / number of actual documents

Thus, high precision value represents that results are substantially more relevant than irrelevant where as high recall value represents that most of the results are relevant. Below is the graph that represents precision Vs Recall for SDPR System.

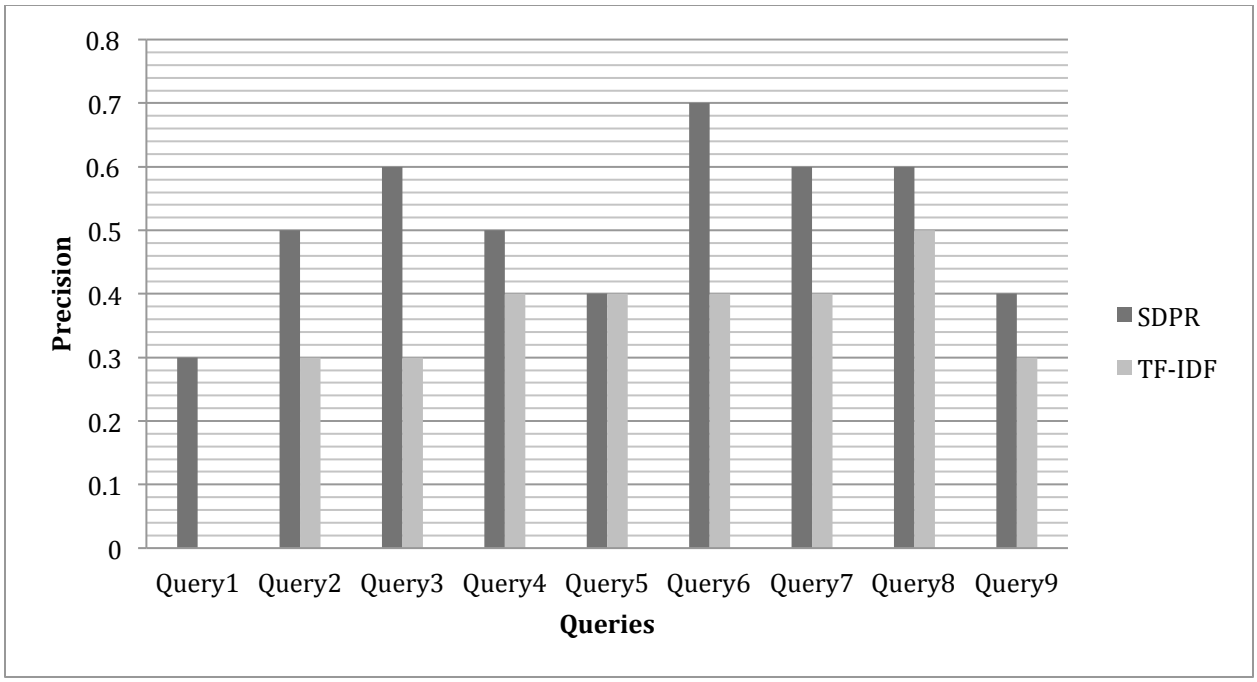

FIGURE 7.1: SDPR Vs TF-IDF Precision

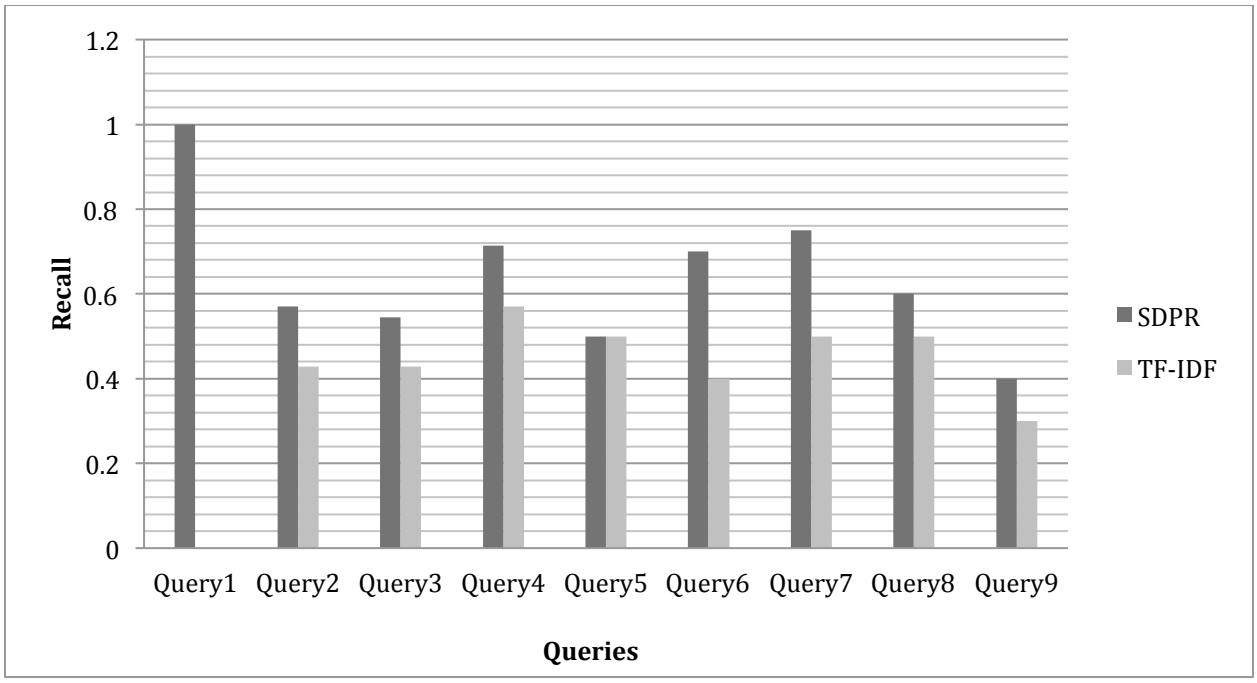

FIGURE 7.2: SDPR Vs TF-IDF True positive rate

7.3 F-Measure or F-Precision Score

F-score considers both precision and recall to consider the relation between both of them. F-measure can be interpreted as the weighted average of the precession and recall, where it reaches its best value at 1.0 and worst value at 0.0. It is defined as follows:

F- score = 2 * precision *recall / (precision + recall)

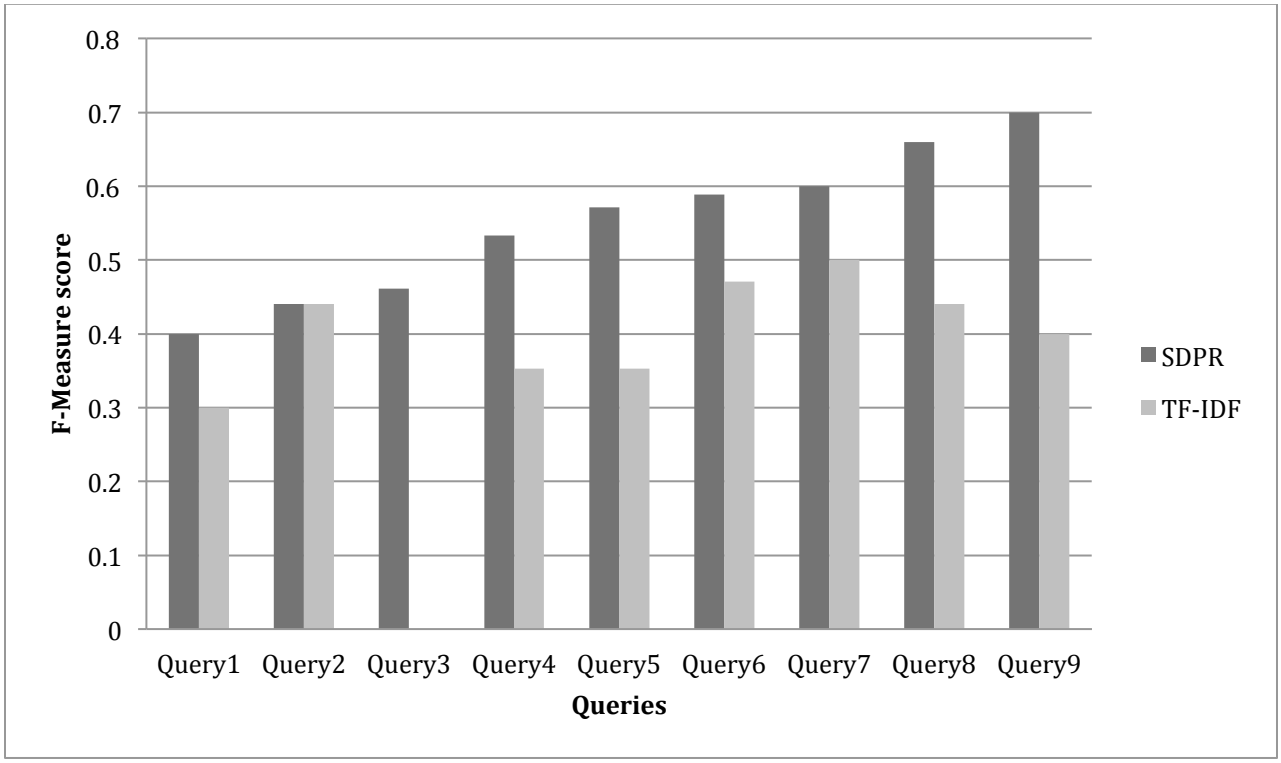

FIGURE 7.3: F-Measure Graph SPDR VS TFIDF

7.4 Individual query Precision Vs Recall graphs

Below is the graph that compares the results that we have obtained from the SDPR algorithm and TF-IDF algorithm. Graph plotted below represents the performance of each of the algorithm for respective queries. We have compared each algorithm with nine queries. Thus, each point in the graph below represents a query. Thus, comparing both of the algorithms on the grounds of precision and recall can lead us to understand actual relevant documents obtained for each of the algorithms for respective queries.

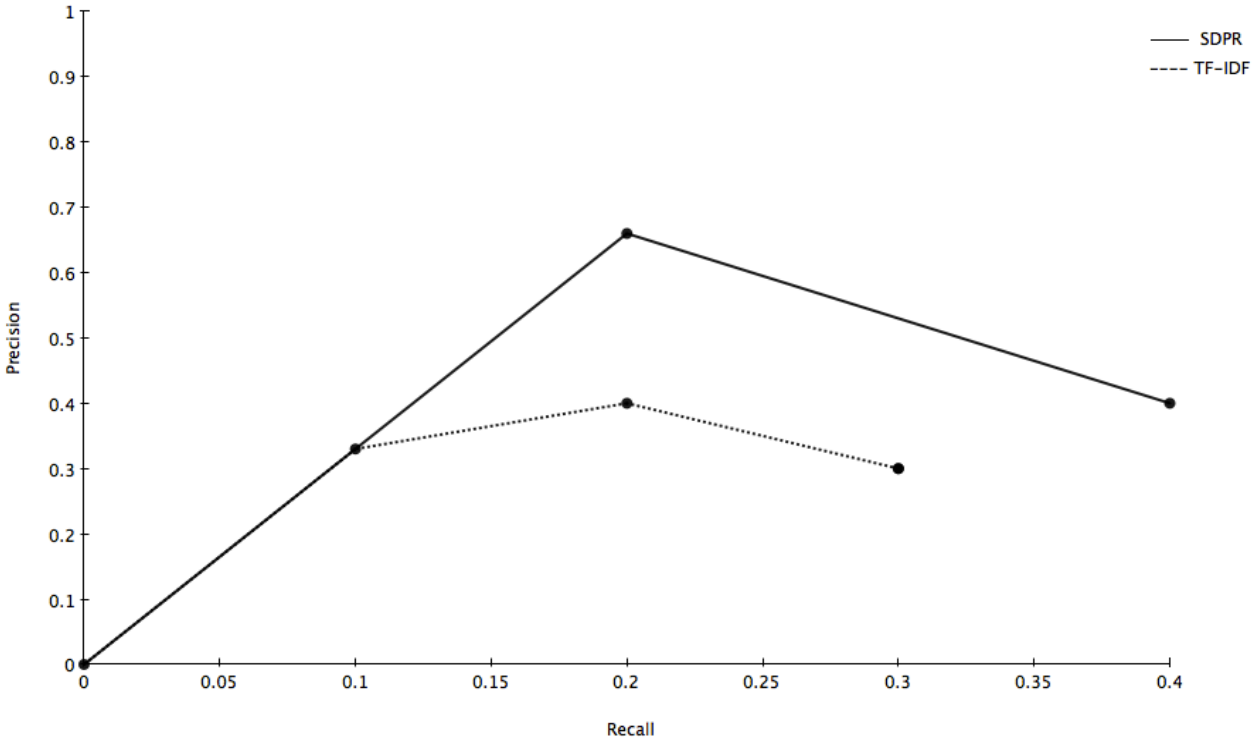

Figure 7.4: Precision Vs. Recall curve for Query1

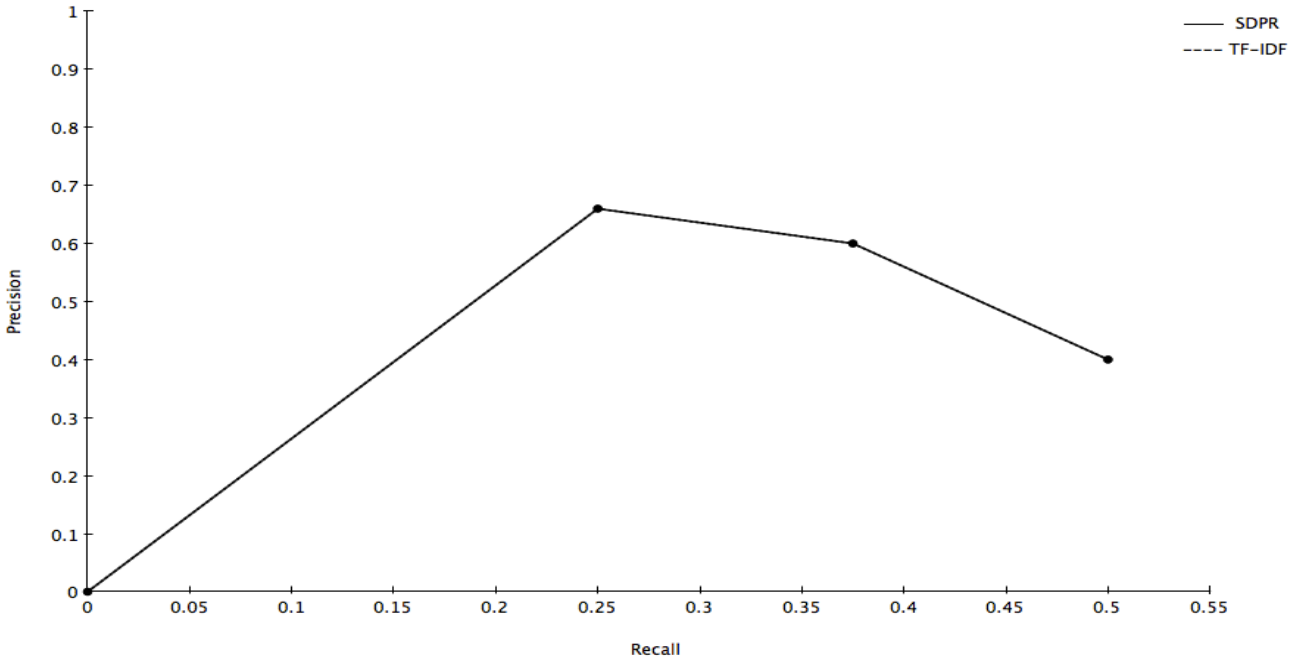

Figure 7.5: Precision Vs. Recall curve for Query2

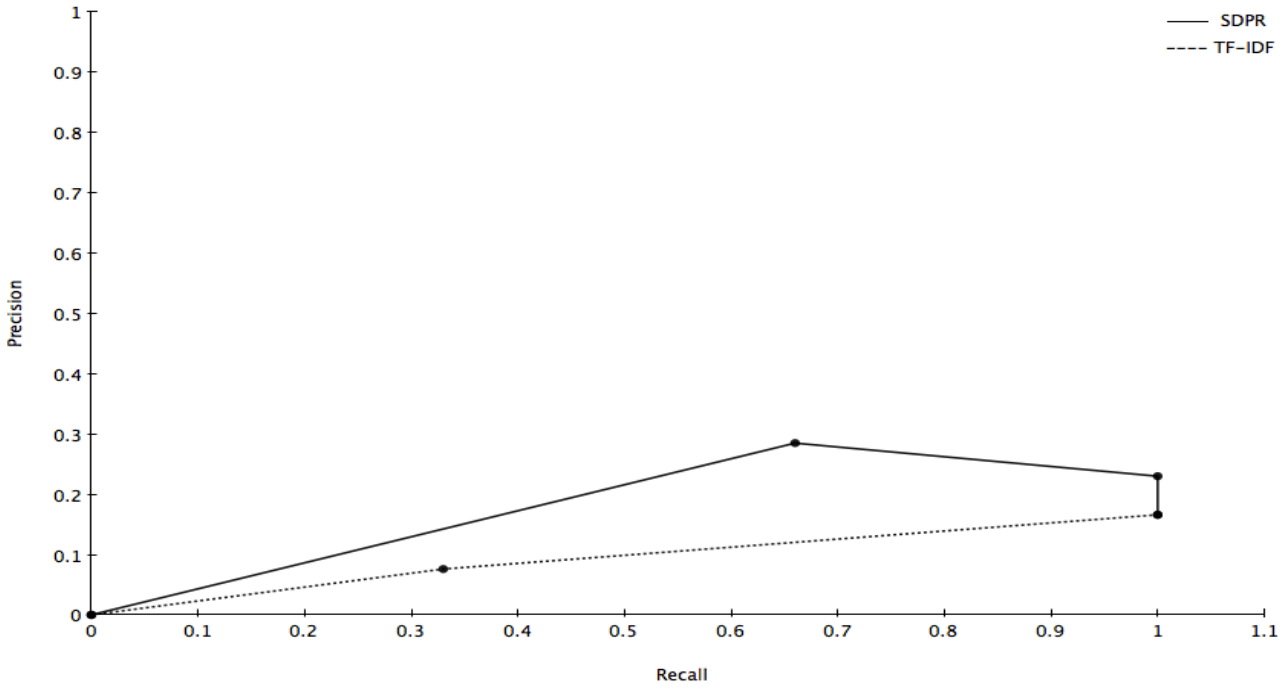

Figure 7.6: Precision Vs. Recall curve for Query3

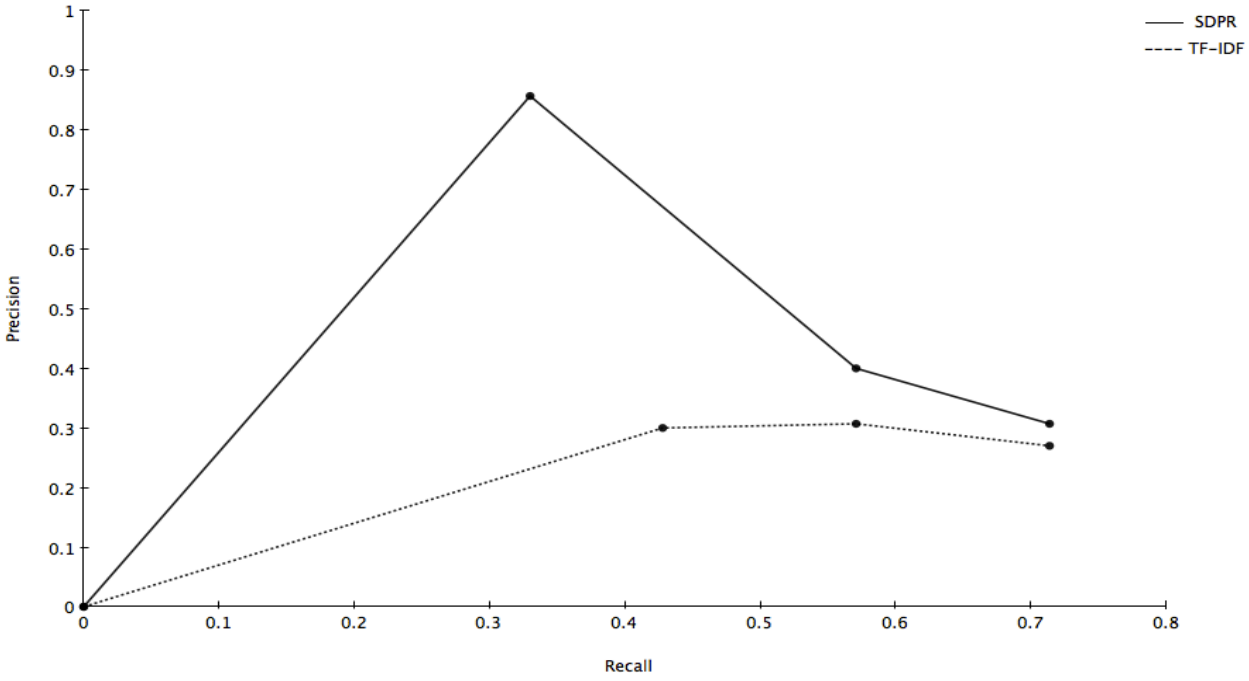

Figure 7.7: Precision Vs. Recall curve for Query4

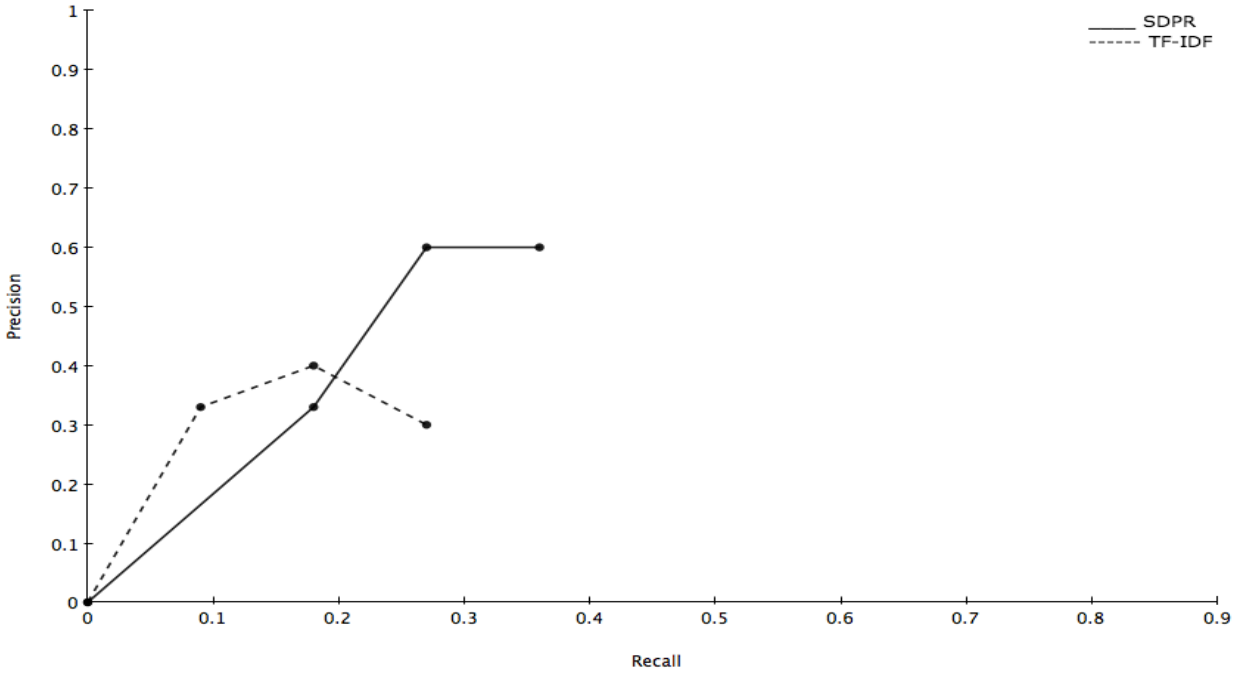

Figure 7.8: Precision Vs. Recall curve for Query5

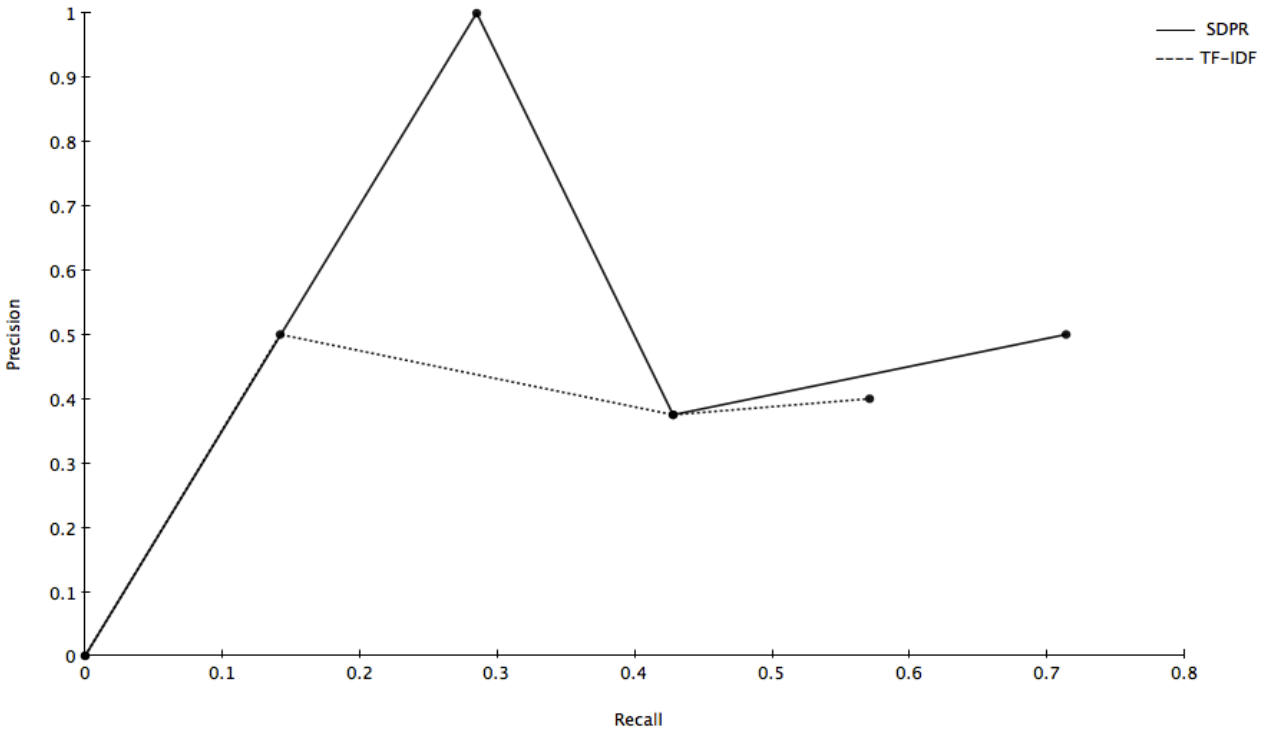

Figure 7.9: Precision Vs. Recall curve for Query6

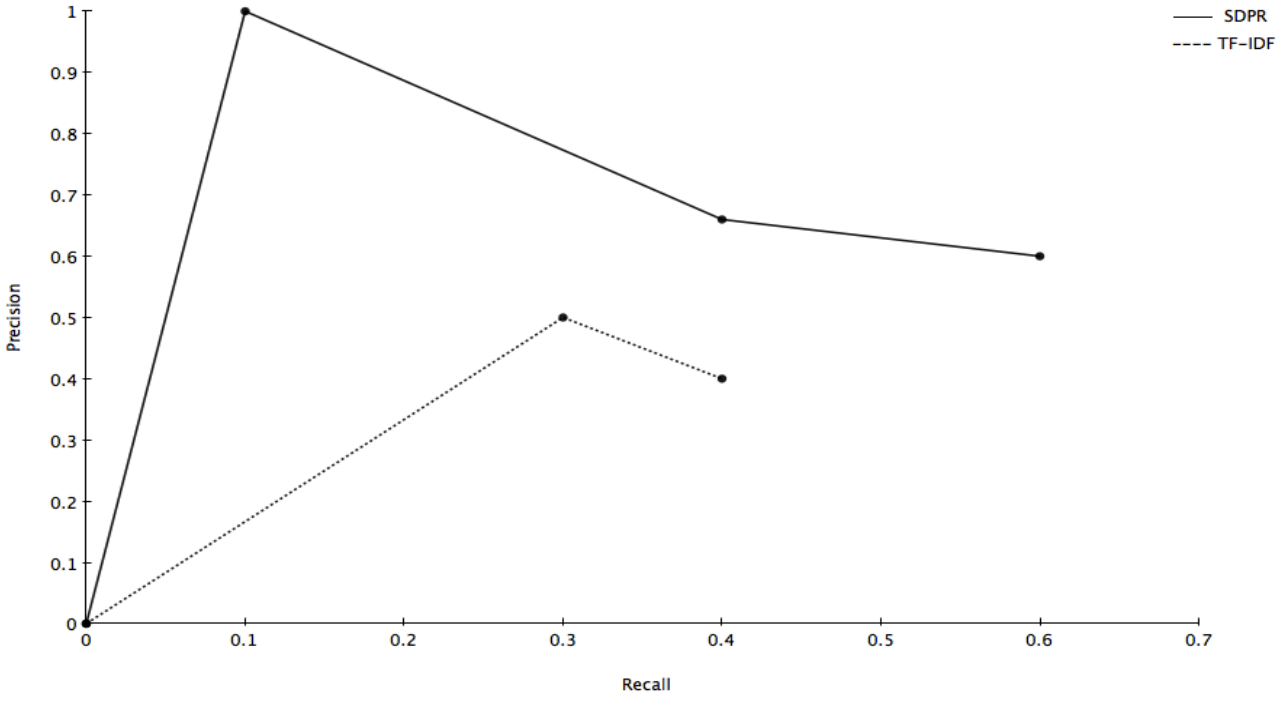

Figure 7.10: Precision Vs. Recall curve for Query7

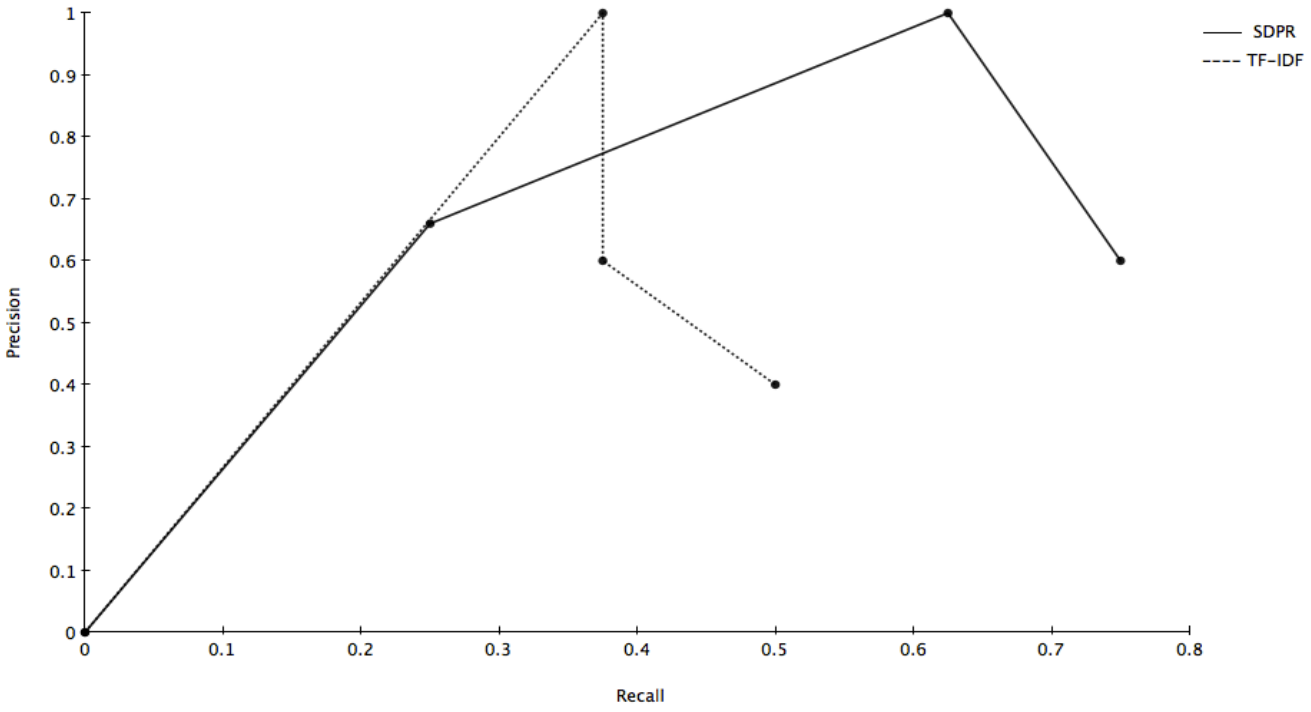

Figure 7.11: Precision Vs. Recall curve for Query8

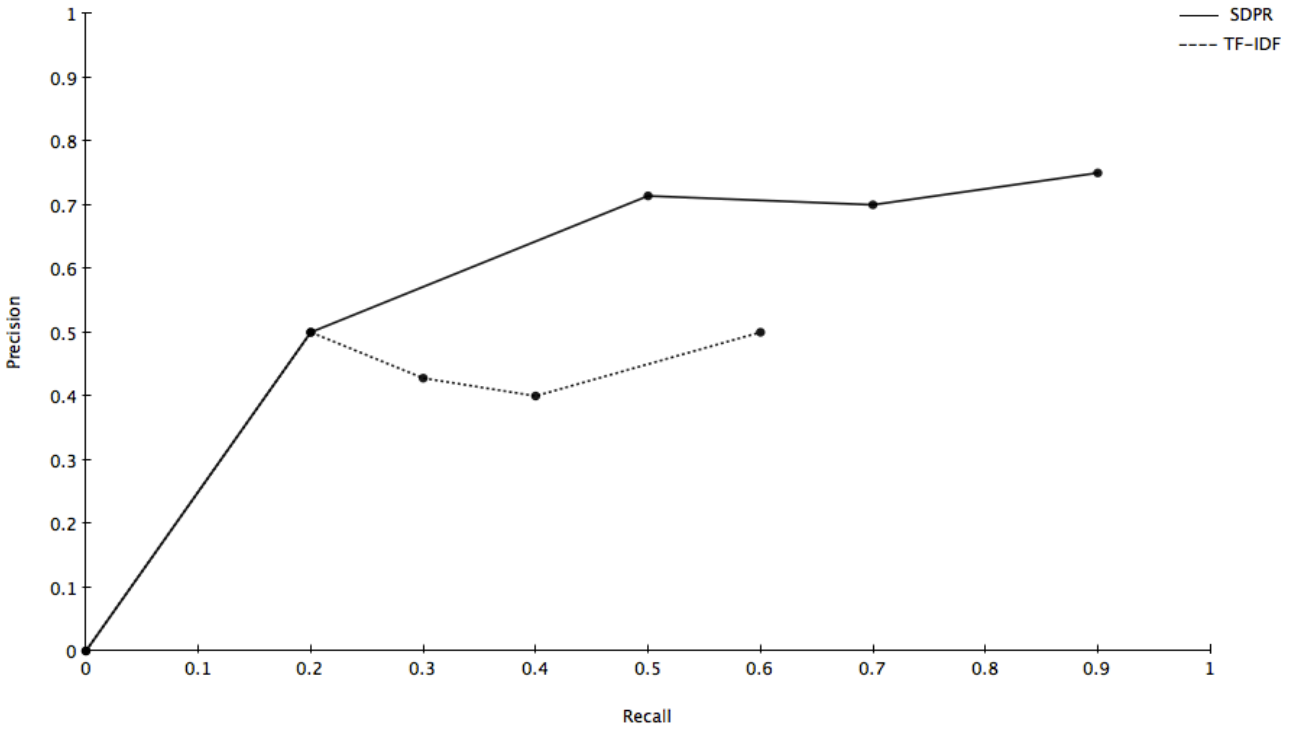

Figure 7.12: Precision Vs. Recall curve for Query9

7.5 False Positive And False Negative

False positive result indicate that the some of the results which appear in the relevancy ranking but actually they are not supposed to be in the result. Whereas, False negative result indicates that, the result appearing in the non-relevancy ranking of the document should be actually present in relevancy ranking. As stated above we tested each of the queries with nine human subjects. Each of the subjects was asked to identify relevant result for each query out of the corpus. We then executed each query with the system SPDF and TF-IDF. We set the threshold of ten documents for each ranked research paper result returned from the systems. Then, Each human subject was again asked to identify the false positive elements in the result set.

We then calculated false positive rate and false negative rate as defined below:

Let, γ be the false positive elements in the result set

λ be the true negative elements in the result

μ be the true positive elements in the result

and ν be the false negative elements in the result

Thus, false positive rate α is calculated as:

$$\alpha = 100 * \frac{\gamma}{\gamma + \lambda}$$

and false negative rate β is calculated as:

$$\beta = 100 * \frac{\mu}{\mu + \nu}$$

We now plot the graph for false positive and false negative for each query:

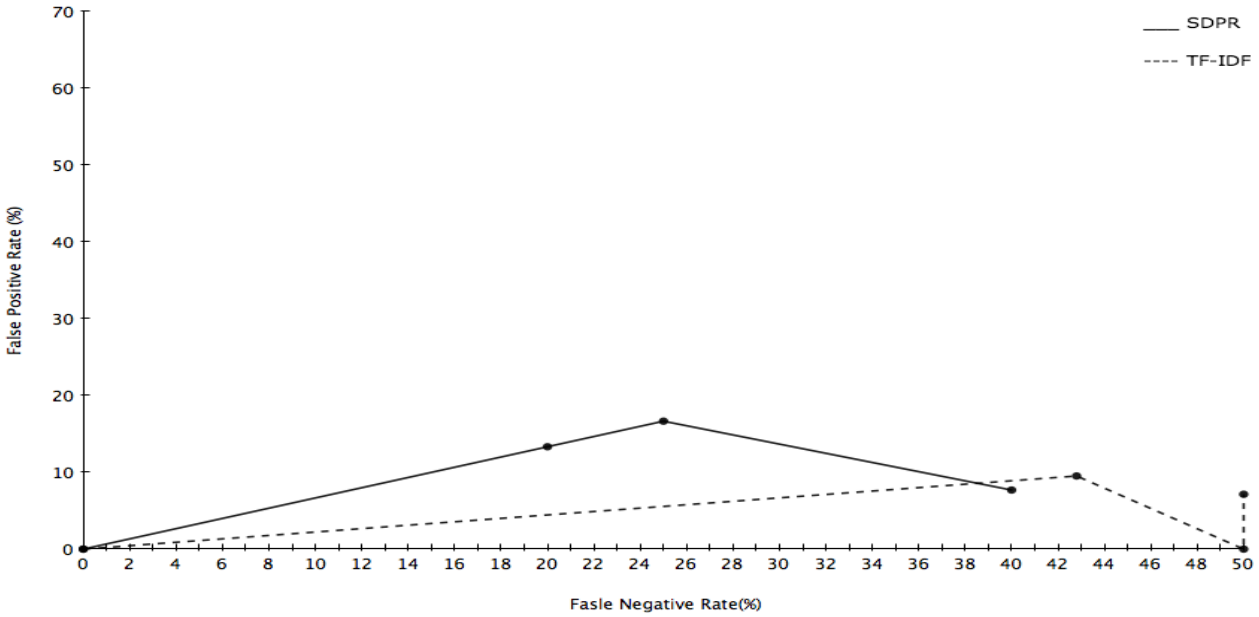

Figure 7.13: Fasle Positive Vs False negative rate for Query1

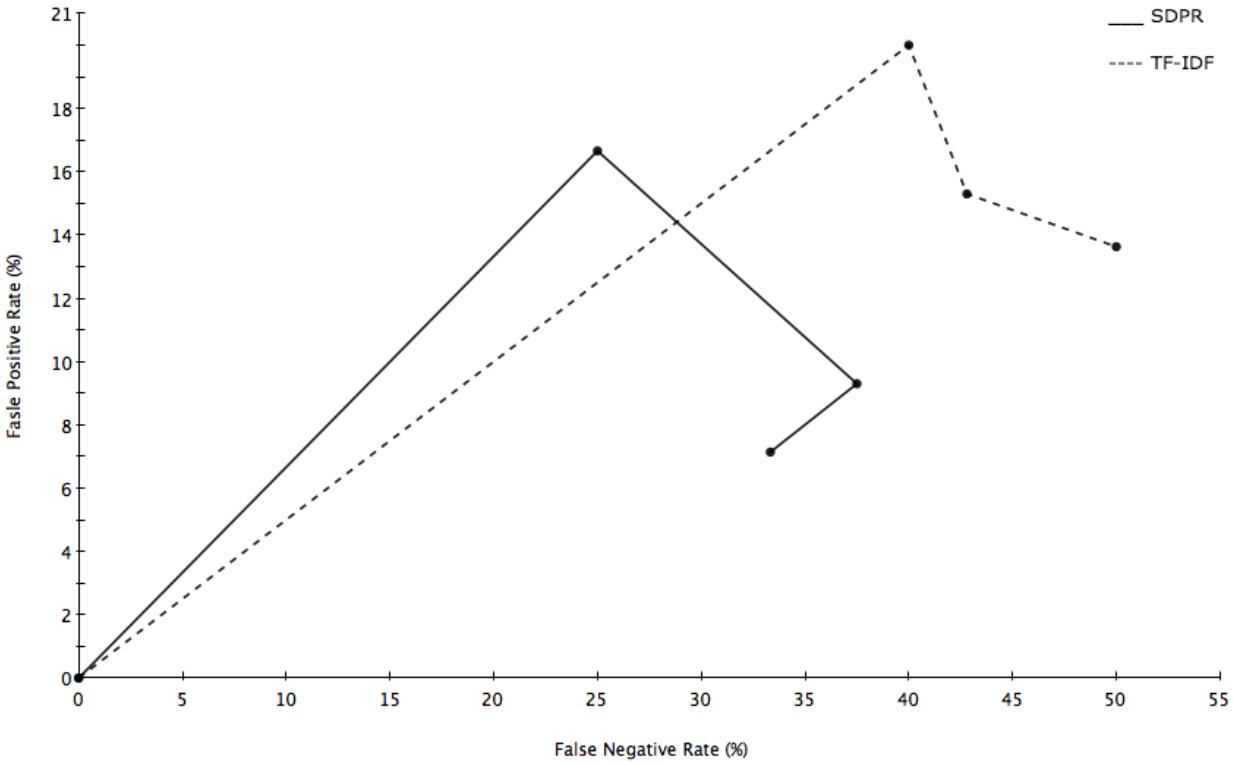

Figure 7.14: Fasle Positive Vs False negative rate for Query2

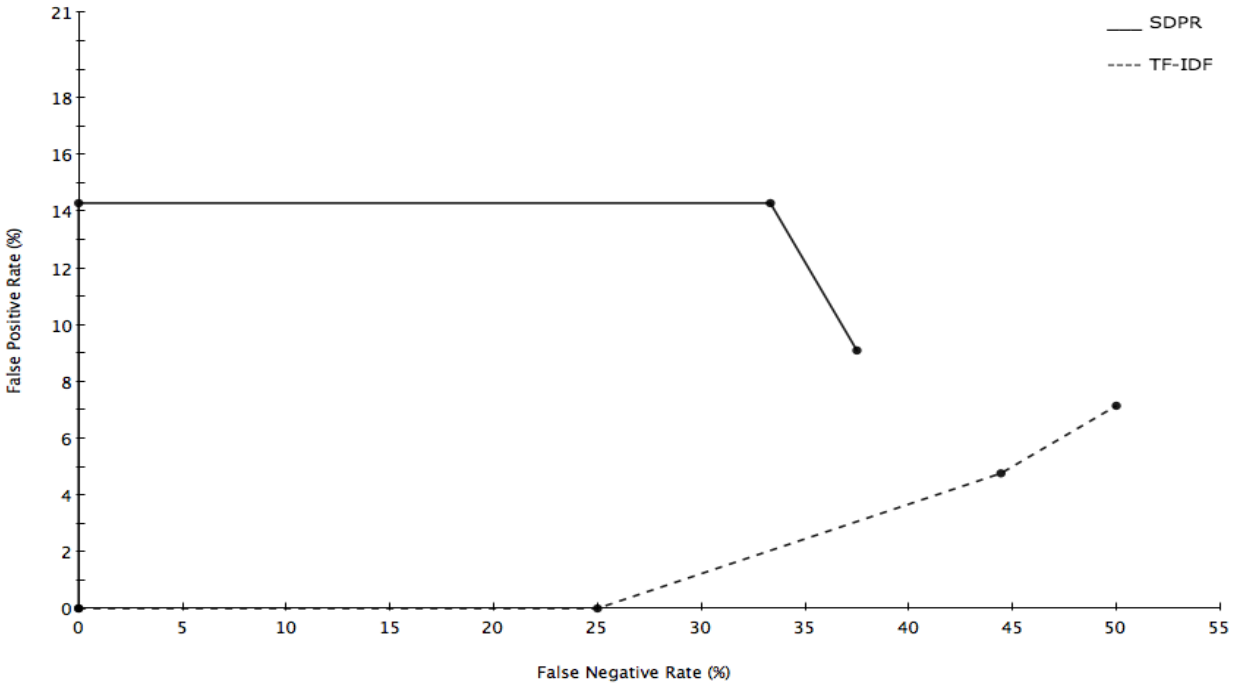

Figure 7.15: Precision Vs. Recall curve for Query4

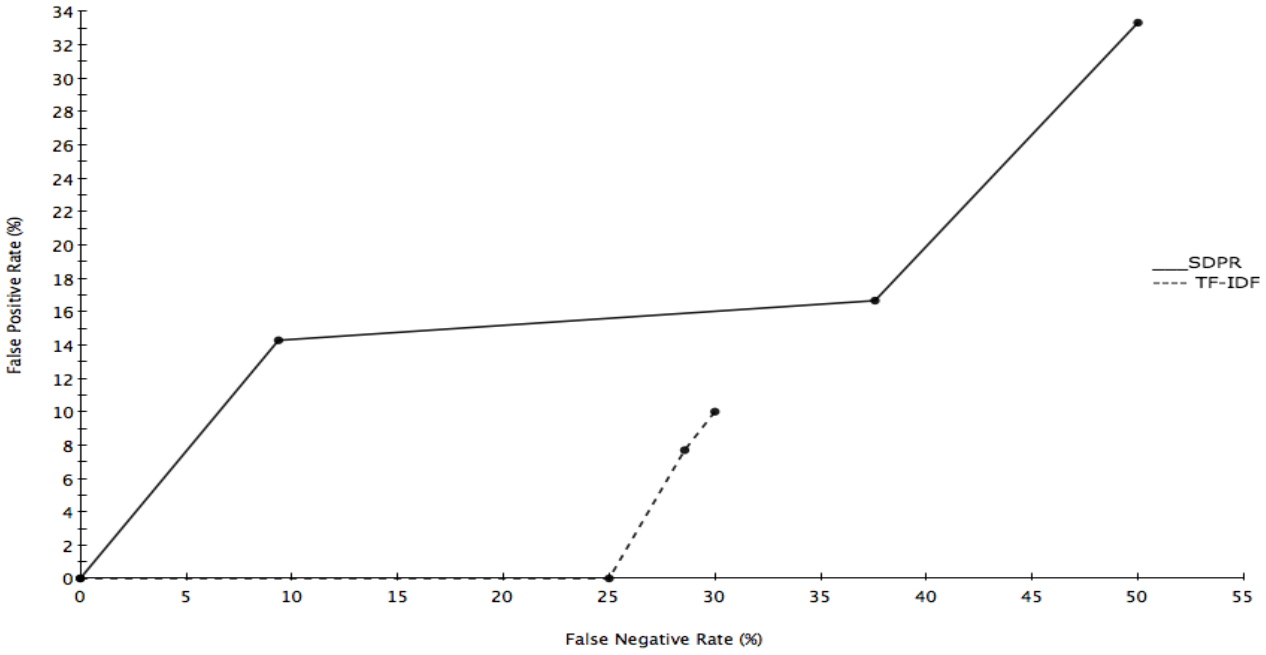

Figure 7.16: Fasle Positive Vs False negative rate for Query5

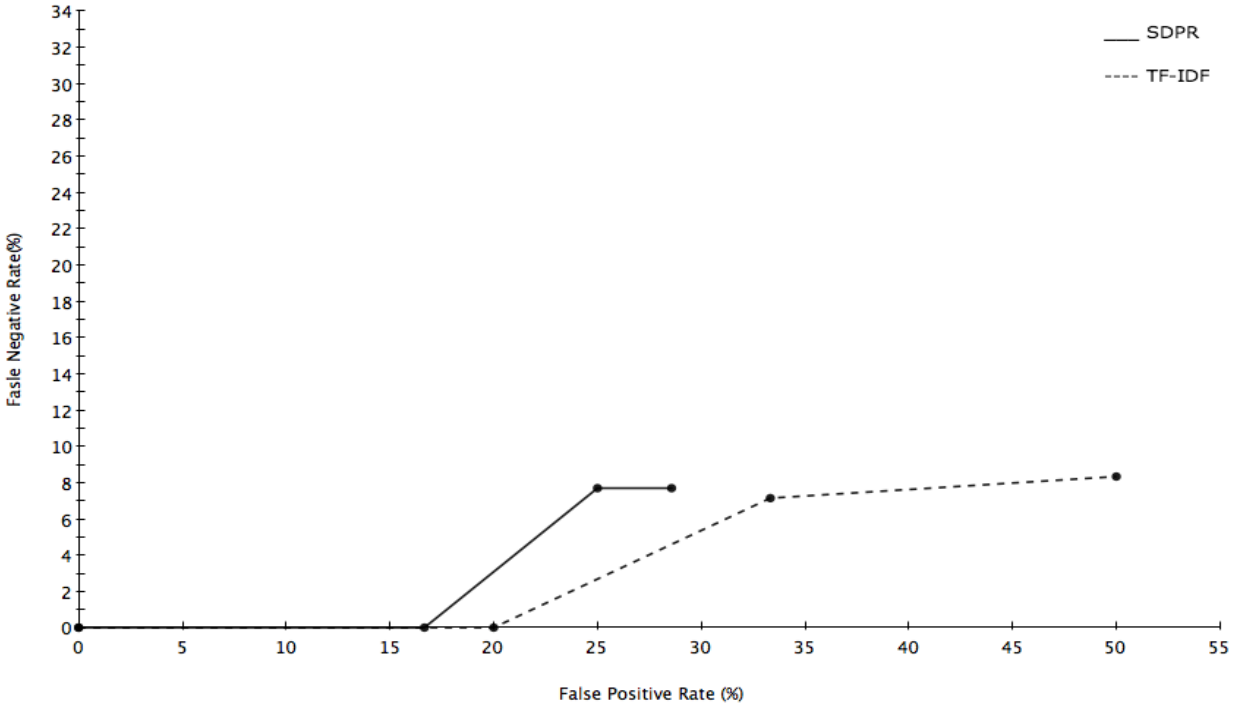

Figure 7.17: Fasle Positive Vs False negative rate for Query6

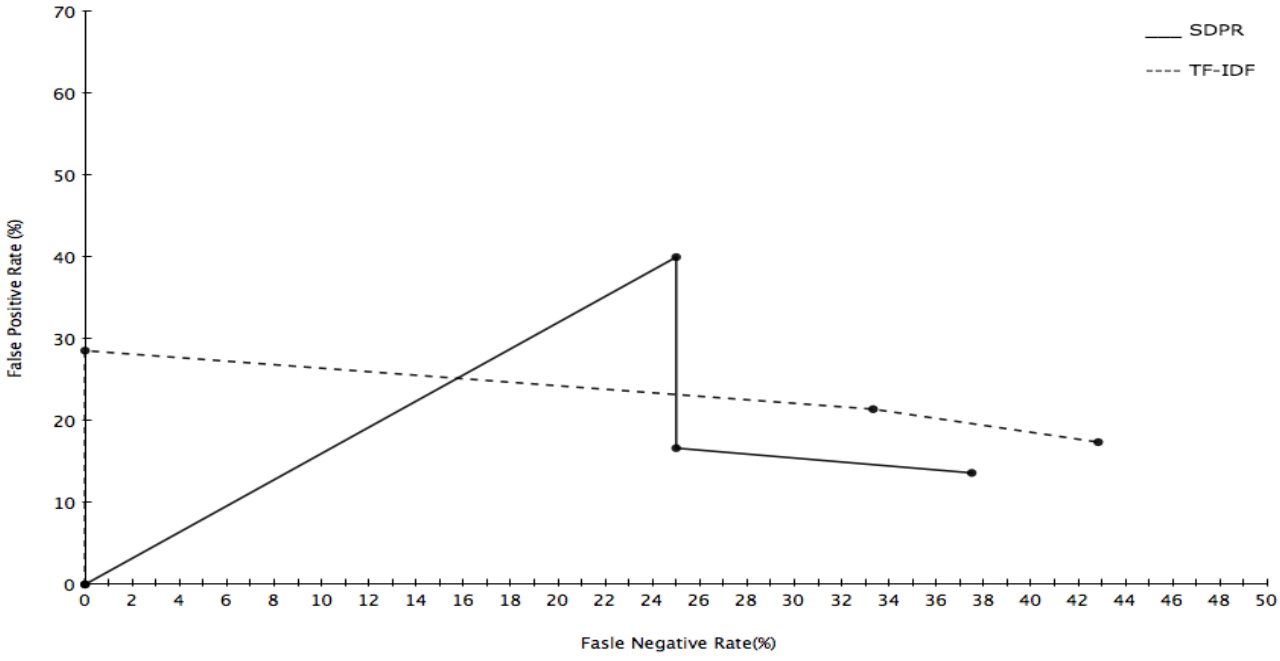

Figure 7.18: Fasle Positive Vs False negative rate for Query7

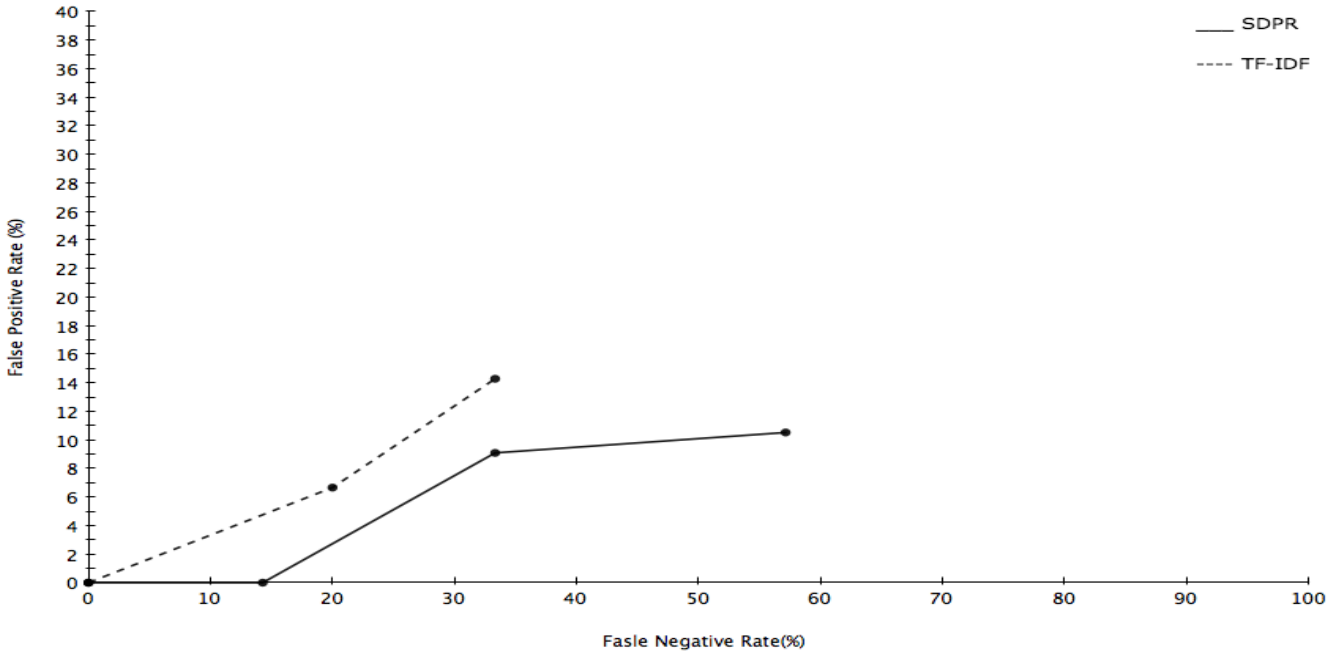

Figure 7.19: Fasle Positive Vs False negative rate for Query8

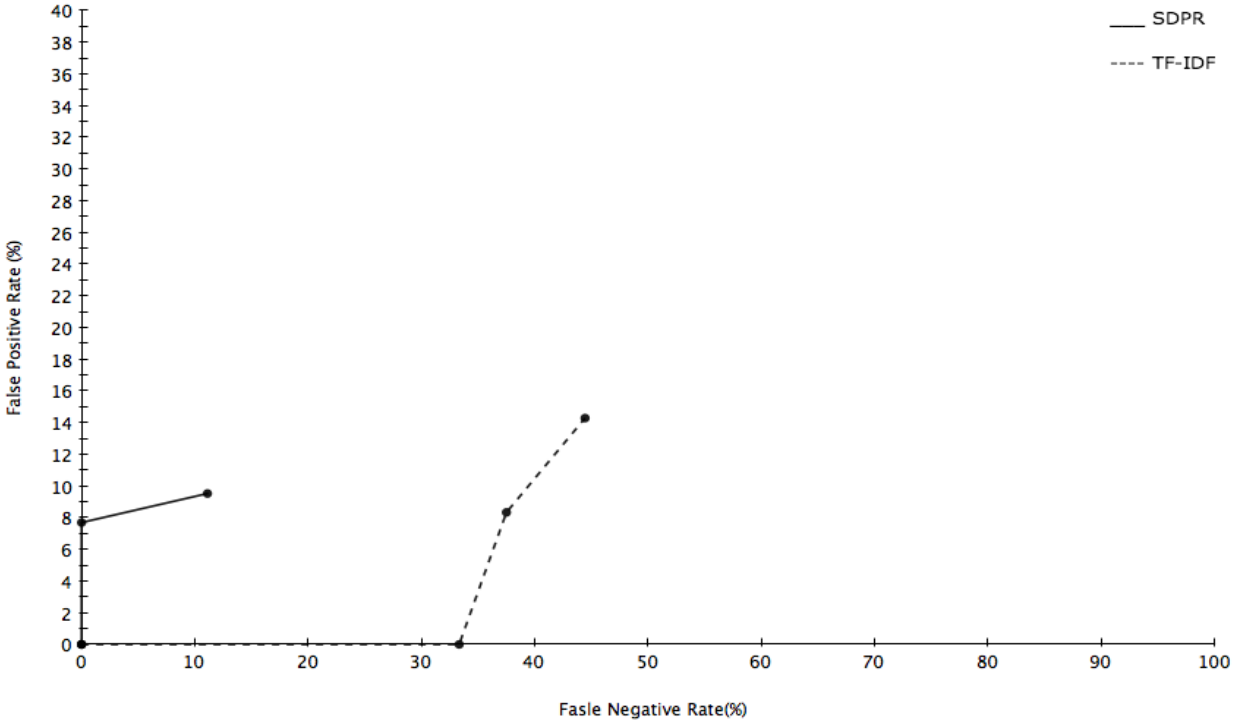

Figure 7.20: Fasle Positive Vs False negative rate for Query9

Queries considered for testing were selected based on the coherence between the words in it. We considered the parameters such as the synonymy, polysemy and anaphora of a word. Anaphora is a type of expression whose reference depends on another referential element. For each of the query we vary threshold to get the quality and accuracy of the query. As seen from the above results considering semantic distance with consideration of term weighting gives out better results than the key word based searching system TF-IDF.

Queries 9,3,4,6, and 7 in the query set are the one that have coherence in the words with an ambiguity. As in the Figure 7.6, 7.7, 7.9, 7.10 and 7.12 the precision and recall is high for low thresholds and is increasing steadily for SDPR representing a better result. This transitively suggests the better query disambiguation in the results for SDPR. As in Figure 7.6 and 7.10, for queries 3 and 7, the precision and recall is high at lower thresholds for SDPR, whereas it is low for TF-IDF i.e. relevant research papers are highly weighted in SDPR result set. Queries 3 and 7 thus show high degree of polysemy. In figure 7.13 and 7.14, it can be noted that false positive rate and false negative rate for these queries is low and thus represents the accuracy of precision and recall for each query.

When the words with acronyms are used as in query 6, the behavior of the result depends on the coherence between other words present in the query and the degree of polysemy of these w.r.t the paper. This can be seen in Figure 7.9. Use of words that have less or none sematic relation to rest of the words in query, as represented in query 5 and 8, results in projection of its dependence on term weight. This can be noted from Figure

7.13 and 7.14. The false negative rate for query 8 is over 40% and false positive rate for query 5 is over 4%, which is higher than TF-IDF. This represents that the system is under performing for these two queries and thus favors keyword based searching in this case.

We observed that acronym or a noun in the query tend not to form synset and thus fail to identify synonymous phrases in the paper resulting in dominance of term weight in result set. Thus, the coherent words in query that provides more information about acronym or noun help in achieving polysemy and synonymy.

Figure 7.5 represents a graph for query 2, which has the words that have synsets with limited number of terms in it. We have considered three terms in this case. As it is observed from Figure 7.13, the False positive for SDPR system for this query is high, thus, in this case it is dominated by term weight more than semantic weighting.

CHAPTER 8

RELATED WORK AND BACKGROUND

Number of research papers published every year is increasing exponentially . Thus, it is a challenge to search thorough all this data efficiently to get to the right document. It is significant that systems should be efficient enough to understand the semantics of the paper and understand the underlying context of the paper. This essay argues that in order to search documents and especially while dealing with research papers its very important to understand the context of the query of users and that of documents by considering semantic relationships between the words as well as considering probability, that certain document is relevant to the query can improve ranking and help in disambiguation of the query.

While considering the challenges as to how the documents should be ranked properly with consideration of the semantics of the paper it is also very important to consider the quality of the paper w.r.t. query. Some of the models fail to take this factor into consideration that highly vary from user to user with their opinion. For example, research paper search system like Google scholar tries to resolve this problem by using 'citations' in the paper as parameter. It tries to see, in how many documents the particular paper is cited; determining the quality of the paper, as the citations for the paper increases so does the quality of the paper. But, while a user tend to search through papers he tend to consider different factors, which can be easily missed out

while considering the citations based ranking of documents. Now let us look into some of the factors considered by a user while searching a document, they are as follows:

1. Perceive underlying context -

   Check what is the domain of the paper. Sometimes just going through abstract can be misleading if the concepts expected are too specific. For example, if user is searching for paper on spread activation models and paper is talking about semantic associations

2. Consideration of content similarity -

   User always tends to check the exact words of the concept what he is looking for in the paper to make the decision.

3. Semantic propinquity of words -

   Not only words but also some of the concepts or phrases user is looking for in the document might not be present, but words or terms meaning the same can be present in document.

Information retrieval has been an intriguing topic for researchers for past many decades, in midst of improving the searching models different approaches were considered like binary search model in which the document is treated as either it matches or not, probabilistic models that work on indexing words principle, adaptive learning, machine learning techniques, semantic models, etc. We would consider some of these papers to go through which have proved to be very significant for this study.

In 1957, Luhn [5]suggested a method to rank a document based on the degree of the words appearing in the document. In this statistical method he considered indexing the

documents and the automatically encoding them. His ranking of the document was based in frequency of the word appearing in the document, which is known as Term-frequency (tf). Further, in 1972, Spärck Jones complemented term-frequency with Inverse document frequency (idf). He suggested in his paper [2] that words, which are repeating in most of the documents, are not useful and thus we need to reduce the weight on such documents. Thus, the frequency of a particular term is checked across all the documents and if its frequency is more its less important while as if it is more, it is considered to be important. In this paper [3] authors have tried to evaluate all the existing systems of indexing which were present at that time and pointed out the inadequacies of these methods and remedies for them. Thus, they went on to propose a novel method for indexing [4], which is based on Singular Value Decomposition (SVD), which they name it as Latent Semantic Indexing. In this paper, they suggest that indexing can be expressed as the density of the object in the space. Thus, by evaluating how far the objects are from each other they can be indexed in different classes or groups for which authors have defined centroid. They use this centroid to calculate the density of that particular class in the space and thus leading to detect the similarity between the documents. In this paper[6], authors implement the partial document ranking which is based on the ranking the documents without going through all the terms in the query at once. So, the documents are ranked as the terms are encountered. Here, authors are trying to evaluate three heuristics for ranking one is L method which was proposed by Smeaton and van Rijsbergen and the other two named as W and SW methods proposed by authors. All the three methods vary ordering of the words while

evaluating and rank the documents. Donna Harmsn studied the factors that affect the ranking of the documents. In his paper[7] he describes the different parameters used for this experiment such as importance of the term in the entire document collection, its significance in a particular document and length of the document. Author concludes that the combinations of different weight measures can make a significant difference in document ranking. In this paper [8], author explains the method of weighting the terms appearing in the query using random walk model. Here positional values of the terms in the documents are considered to compute the inverse document frequency and then terms in the query are assigned with trained weights. This method is based on the graph based ranking algorithm where terms in the documents are represented with graph of that document. In IR indexing is widely studied area and author of this paper [9] proposed a method for indexing the words in the documents using probabilistic model. Author suggests that better indexes can be formed by defining measure of *indexability* that reflects the importance of the words in the documents relatively. According to the model each word represents a concept, which is treated in two degrees. The words are divided in two degrees using the Poisson distribution with help of the γ_1 and γ_2 that represents class I and class II which are in turn defined by relevant request for information on the concept and k number of tokens of the words. In this paper [10], authors explain a method for ranking the documents using semantic relationships. Here, identifying the significance of the metadata in industry and academia, authors create semantic associations between all the entities on web by discovering complex relations in RDF data. This can be used for flexible ranking, which

can be used to discover relationships in semantic web that are more relevant and interesting.

Thomson Paul, in his paper [11], peeks back into Maron and Kuhns paper[12] and for the first time coined the probabilistic method of document ranking. They use weight, as the probability of the query matching to the document instead of the two value indexing. Thus the output of their model was not the matched documents with the query index but set of ranked documents in descending order. Authors named this as 'probabilistic ranking'. The probability here is calculated as $P(D_i \mid I_j)$ where I_j is the topic requested by the user in the query and D_i is the event that document is relevant to the users topic. Further, S.E. Robertson, in his paper[13] did a detailed analysis of the probability theory involved in finding the relevance between document and the query. In his paper he introduced different justification for using probability in IR, which are traditional measures of effectiveness and decision theory. Also, he tried to explain different ranking principle to justify probability of satisfaction for ranks of the document. In this paper[14], authors have explained the ways in which two Poisson models for term frequencies can be used for probabilistic information retrieval by introducing different variables in them related to term frequency in documents, document length and within-query term frequency. In this paper [15], authors explain the development of probabilistic model with the experiment. They have used TREC materials for experimentation. The paper is divided in two parts, where in first part authors forms the foundation of probabilistic models and in second part further development of probabilistic model with extensive study and experiments is done. In

this paper [16], author have emphasized on the significance of the term of the query while retrieving information. He has introduced a modeling approach, which he named as notion of importance. The model takes into consideration term significance with help of probabilistic approach, stop words, mandatory terms and coordination level ranking. In this paper [17], authors point out the factors like acquiring and utilizing knowledge, which are needed to realize semantic web vision. Also, they try to briefly discuss scalability problem for search systems using three classes viz., search/browsing, integration and analytics. Authors from Microsoft explained in their paper[18] a method using clicks to personalize the search. Here, authors define class and the probability of user query to be in certain class to weight particular page, thus ranking the pages. Authors of this paper[19], have studied similar concept as in [18] but have used different approach. Here, authors propose a model named as conditional random field (CRF), which uses neighboring queries and corresponding URLs that are clicked in the log to weight particular page. In another paper [20] based on similar theory for click based ranking model where authors introduce with novel approach of context aware query suggestion. The model works in two steps, where in first step of offline model it tries summarize queries by clustering and then building a suffix tree for query suggestion model. In second step of online model user`s context is captured to suggest the queries. While ranking documents it is very important to pay an attention towards ranking and that is what is studied in this paper[21] by authors, where to address the issue of expensive and time consuming procedure of self learning in many systems they have introduced a novel method pool-based active learning which represent label

ranking function by considering both label decomposition and constraint classification technique. In another paper [22] based on ranking modeling, authors tries to argue that ranking/sorting systems which generally are solved using classification can also be solved by considering inherent structure of the data.

In this paper [23], authors have described the method of query association for document retrieval. Here, they use a query, which is highly similar to the document as the descriptor for that document and further used to rank for other queries. Authors of this paper [24] argued that vector space model fail to consider the spatial information and in order to bridge this gap they have introduced method named as Fourier Domain Scoring (FDS). Here, they score each document through five steps i.e. Collect words into spatial bins, Create inverted index, Perform pre-weighting, Perform Fourier transform and calculate document score. In this paper[25], authors have introduced a method of file signature where files having same term frequencies are assigned with same signature. Hashing the files with term frequencies creates multiple file signatures. According to authors of this paper [26], in the extended Boolean Models, belief revision can form theoretical framework for document ranking. Here authors have proposed a similarity measure for a model based on proportional logic for information retrieval, which is similar to p-norm and thus have similar properties and behavior. Authors of this paper [27] apply variant of perceptron algorithm using selective committee averaging. This algorithm tries biasing the final solution to maximize the arbitrary rank based metrics. In this paper[28], authors describe a method to rank document using predefined ontology (ODP) of semantic user profile and then combining user profile

sub graph with documents super graph. Kyung-Soon Lee and et al. explained in their paper[29] the model they developed, which tries to retrieve documents using inverted file method and analyze the terms in documents using cluster analysis. LSI method can be enhanced using the differential term weighting and relevance feedback that was done by Susan T. Dumais in her paper[30]. Moving on to semantic search systems, authors of this paper [31] approached ranking problem using SemRank model, which is a blend of semantic ranking, and modulative search where users can vary the search modes to see the changes in result. Also, authors explain the infrastructure that supports SemRank used in SAARK system. Chengxiang Zhai and John Lafferty proposed a model in their paper[32], which is based on feedback documents. They use two strategies to update the query language model based upon the feedback documents out of which one is generative probabilistic model and other is minimization of the KL-divergence over feedback documents. Evaluation of the IR methods was done in their paper[33], where authors describe two methods of evaluation of which one is P-R curve and average precision computation using recall bases of documents of various degrees of relevance and the other two novel measures using the retrieval result up to a given ranked position. Authors of this paper [34] discuss their experience of building an adaptive technique to build empirically defined, frequency weighed index. When it comes to information retrieval human preferences play an important role. Thus, authors of this paper [35] have applied machine-learning technique to predict variables of ordinal scale, which is named as ordinal regression. They have used an approach, which maps objects to scalar utility values securing transitivity and asymmetry between

objects to ranks mapping. In this dissertation author have views the retrieval as an evidential process where sources of evidence of documents and query content are used to predict which documents are related to the query. In this paper, which is based on probabilistic document indexing [37], authors approach using data collected from set of queries to derive relevance feedback data. The approach is based on three new approaches, which are abstraction from specific terms and documents, flexibility in representation to allow including new knowledge base and estimating index weight using probabilistic learning and classification. Our work is very much related to this paper[38], where authors explain a ranking system that they are developing for testing ranking of documents named as ORank. Here, authors are approaching the problem by combining conceptual, statistical and linguistics of document, expanding the query, using and modifying spread activation algorithm for weighting relationships and allowing variable document vector dimensions. Authors of this paper [39] propose two generic text summarization methods where one uses standard IR methods while as second uses latent semantic analysis to find semantically important sentences for summarization. In this paper, authors have approached problem with three models. The first is generalized version of Maron and Kuhns model which authors named as Binary Independence indexing, second is based on assumptions that each indexing has its correctness to which probabilities relate named as retrieval-with-probabilistic-indexing (RPI) model and the third is descriptors for indexing using controlled vocabulary named as Darmstadt indexing approach (DIA). Authors of this paper [40] introduce a new method using the percentage of relevant documents to rank query-

specific clusters. Here authors use information that is induced by the documents associated with the clusters.

Authors of this paper [41] tried to classify document by introducing combination of differential document vectors and DLSI (differential latent semantic indexing) spaces. Relationships of the documents are one way to retreieve the documents and this is what is considered by Jaroslaw Balinski and Czeslaw Daniowicz in their paper [42] where they use relate documents using any hyperlinks or text in them and use intuition that close documents should not weight differently. In this paper [43]authors creates blocks out of documents using text-tiling algorithms, then they are assigned with scores using manifold algorithm and finally document is ranked by fusing the scores of the document. While ranking the documents every expert emphasis on different aspects of documents, thus, authors of this paper [44] proposed a method to combine all the results through these systems to achieve higher performance. Taeho jo introduced in his paper [45] studied the semantic similarity between the strings with help of introducing three different operations i.e. semantic similarity, semantic similarity average and semantic similarity variance. While ranking documents algorithms tend to keep different vectors of intermediate scores, which cannot be avoided, in rich document ranking systems. Thus, Taher H. Haveliwala in his paper [46] proposes lossy encoding schemes, which are based on scalar quantization algorithms to encode the auxiliary information. Lexical cohesion is an important property of the text, especially when it comes to information retrieval. Thus, the authors of this paper [47] have studied the degree of lexical cohesion between the context of query terms` and the documents using

the lexical –semantic relations that exist between there collocates. Authors of this paper [48] have tried to rank documents using hybrid method i.e. using vector model to downsize the number of documents that are related to query and then have used conceptual graph to represent documents. Our system uses spread activation technology for finding semantic relationships, which is explained by F. Crestani in his paper [49] that describes about the ways this technology, can be used for Information Retrieval. Also, further in the paper, author has explained the structure and working of spread activation model. Soni Dhanni in her paper [50]has explained the semantic relations in Wordnet and ways to access it using RitaWordnet API.

CHAPTER 9

CONCLUSION

We presented the system SDPR: an efficient search framework for ranking research papers. We discussed the three features SDPR has to offer. Firstly, while ranking the research paper it's very necessary to address the significance of the words in the query. The term weight decisions are very much dependent on the context of the document and the significance of the words with a particular document. Thus, we came up with the system to identify significant words from the query. The second feature of the system is the semantic distance calculator. This deals with weighting the relationships between the words of the query and the document. We start with identifying the context of the document followed by using the words representing context to find document`s relatedness with the query. Third, to endorse the semantic ranking with the weighting the significant query terms present in the document, we presented weight calculator.

Currently the system weights the coherence between the words in the query and that of the research papers using the semantic distance between the words. Semantic distance calculator helps in identifying the context of the document. However, different factors can be taken into consideration in order to identifying the context such as keyword tags in the paper and personalization. Tags in the paper can be used to create a graph where each node can point to a different set of corpus. Word classifier can be extended o

categorize the queries based on the terms in it. Each query can be redirected to certain set of research papers to be processed and ranked with SDPR framework.

SDPR framework uses DBLP ontology as a data repository equipped with lots of properties that can be used for personalization of the query result. We can classify the research papers according to the domain and use the tag graphs to redirect queries to certain domain research papers. With the improvement in the indexing system for SDPR we would be able to use the semantics of DBLP ontology to discover the similarity between the query and research paper. We are using ontology as the data repository, but we would like to consider other databases like MySQL or MongoDB.

We would like to extend the architecture to take multithreading into consideration. We can group the worker of threads into two. One group would be responsible for the computations of semantic distances between the query words and document, as this is a heavy operation for the system. The other group of worker threads would be responsible for computations of term weight. We can use map reeducation methods to distribute the data over different servers as we consider larger data sets for testing and add all the processed ranks together to get final ranked document of research papers.

BIBLIOGRAPHY

[1] Essential Science Indicators SM from Thomson Reuters – "http://archive.sciencewatch.com/about/met/"

[2] K. Spärck Jones, 'A statistical interpretation of term specificity and its application in retrieval', Journal of documentation, vol. 28, no. 1, pp. 11-21, 1972.

[3] G. Salton and C. S. Yang, ' On the Specification of Term Values in Autoatic Indexing', Department of Computer Science, Cornell University, Ithaca, New York, 14850, USA, TR 73-173, 1973.

[4] G. Salton, A. Wong, and C. S. Yang, 'A vector space model for automatic indexing', Communications of the ACM, vol. 18, no. 11, pp. 613-620, 1975.

[5] H.P. Luhn, "A statistical approach to mechanized encoding and searching of literary information", IBM Journal Research and Development, Vol. 1, pp. 309-317, 1957

[6] Wai Yee Peter Wong and Dik Lun Lee " Implementations of Partial Document Ranking Using Inverted Files "Department of Computer and Information Science, Ohio State University, 2036 Neil Ave, Columbus, Ohio 43210, U.S.A. May 1992

[7] Donna Harmsn "An Experimental Study of Factors Important in Document Ranking" Lister Hill National Center for Biomedical, Communications National Library of Medicine, Bethesda, Maryland, 20209, 1986

[8] Arif, A. "Information Retrieval by Modified Term Weighting Method Using Random Walk Model with Query Term Position Ranking", International Conference on Signal Processing Systems, 2009

[9] Stephen P Harter "A Probabilistic Approach to Automatic Keyword Indexing "Journal of the American Society for Information Science (pre-1986); Sep/Oct 1975; 26, 5; ABI/INFORM Global pg. 280

[10] Boanerges Aleman-Meza, Christian Halaschek-Wiener, I. Budak Arpinar, Cartic Ramakrishnan, and Amit ShethRanking "Complex Relationships on the Semantic Web", IEEE Internet Computing, vol. 9, issue 3, May/June, 2005

[11] Paul Thompson "Looking back: On relevance, probabilistic indexing and information retrieval" Information Processing and Management 44 (2008) 963–970

[12] Maron and Kuhns "On Relevance, Probabilistic Indexing and Information Retrieval" Journal of the ACM (JACM), Volume 7 Issue 3, July 1960, Pages 216 – 244

[13] S.E. ROBERTSON "THE PROBABILITY RANKING PRINCIPLE IN IR", Journal of Documentation, Vol. 33 Iss: 4, pp.294 – 304, 1977

[14] S.E. Robertson and S. Walker "Some Simple Effective Approximations to the 2–Poisson Model for Probabilistic Weighted Retrieval" conference on Research and development in information retrieval Pages 232 – 241, ACM SIGIR (1994)

[15] Karen Sp¨arck Jones “Document retrieval: shallow data, deep theories; historical reflections, potential directions” LNCS 2633, Berlin: Springer, 2003, 1-11, ECIR 2003

[16] Djoerd Hiemstra “Term-Specific Smoothing for the Language Modeling Approach to Information Retrieval: The Importance of a Query Term” August 11-15, ACM 1-58113-561-0/02/0008, SIGIR (2002)

[17] Amit Sheth and Cartic Ramakrishnan "Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis" Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 2003

[18] Paul N. Bennett, Krysta Svore and Susan T. Dumais “Classification-Enhanced Ranking” ACM 978-1-60558-799-8/10/04, WWW 2010

[19] Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang "Context-Aware Query Classification" ACM 978-1-60558-483-6/09/07, SIGIR 2009

[20] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He "Context-Aware Query Suggestion by Mining Click-Through and Session Data" ACM 978-1-60558-193-4/08/08, KDD 2008

[21] Klaus Brinker” Active Learning of Label Ranking Functions” International Conference on Machine Learning 2004

[22] Shyamsundar Rajaram, Ashutosh Garg, Xiang Sean Zhou, Thomas S. Huang "Classification Approach towards Ranking and Sorting Problems"

Lecture Notes in Computer Science Volume 2837, 2003, pp 301-312, Machine Learning: ECML 2003

[23] Falk Scholer and Hugh E. Williams "Query Association for Effective Retrieval" ACM 1-58113-492-4/02/0011, CIKM 2002

[24] Laurence A. F. Park Kotagiri Ramamohanarao and Marimuthu Palaniswami "Fourier Domain Scoring: A novel document ranking method" IEEE Trans. Knowledge and data Engineering, 2002

[25] Dik Lun Lee and Liming Ren "Document ranking on weight-partitioned signature files" Technical report HKUST-CS94-39, 1994

[26] David E. Losada and Alvaro Barreiro "Using a Belief Revision Operator for Document Ranking in Extended Boolean Models" ACM 1-58113-096-1/99/0007, SIGIR 1999

[27] Jonathan L. Elsas,Vitor R. Carvalho and Jaime G. Carbonell "Fast Learning of Document Ranking Functions with the Committee Perceptron" WSDM 2008

[28] Mariam Daoud, Lynda Tamine, and Mohand Boughanem "A Personalized Graph-Based Document Ranking Model Using a Semantic User Profile" UMAP 2010, LNCS 6075, pp. 171–182, 2010

[29] Kyung-Soon Lee, Young-Chan Park, Key-Sun Choi "Re-ranking model based on document clusters" Information Processing and Management 37 (2001)

[30] Dumais, Susan. "Enhancing performance in latent semantic indexing (LSI) retrieval." (1992).

[31] Kemafor Anyanwu,Angela Maduko and Amit Sheth "SemRank: ranking complex relationship search results on the semantic web" WWW 2005

[32] Chengxiang Zhai, John Lafferty “Model-based Feedback in the Language Modeling Approach to Information Retrieval” CKIM 2001

[33] Kalervo Järvelin & Jaana Kekäläinen "IR evaluation methods for retrieving highly relevant documents" SIGIR 2000

[34] George W. Furnas “Experience with an adaptive indexing scheme” Human Factors in Computing Systems CHI 1985

[35] Ralf Herbrich and Thore Graepel, kalus Obermayer “Regression Models for ordinal data: A Machine learning Approach” Technical University of berlin 28/29, 10587, 1999

[36] Howard Turtle “Inference networks for document retrieval” dissertation, university of Massachusets, 1991.

[37] Norbert Fuhr, TH Darmstadt and Chris Buckley “A Probabilistic Learning Approach for Document Indexing” ACM Transactions on Information Systems, Vol 9, No. 3, Pages 223-248, July 1991

[38] Mehrnoush Shamsfard, Azadeh Nematzadeh, and Sarah Motiee "ORank: An Ontology Based System for Ranking Documents" IJCS VOLUME 1 NUMBER 3 2006

[39] Yihong Gong, Xin Liu "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis" SIGIR 2001

[40] Oren Kurland, Eyal Krikon “The Opposite of Smoothing: A Language Model Approach to Ranking Query-Specific Document Clusters” Journal of Artificial Intelligence Research archive Volume 41 Issue 2, May 2011

[41] Liang Chen, Naoyuki Tokuda, Akira Nagai "A new differential LSI space-based probabilistic document classifier" Information Processing Letters 88 203–212 (2003)

[42] Jaroslaw Balinski, Czeslaw Daniłowicz “Re-ranking method based on inter-document distances” Information Processing and Management 41 759–775 (2005)

[43] Xiaojun Wan, Jianwu Yang and Jianguo Xiao” Towards a unified approach to document similarity search using manifold-ranking of blocks” Information Processing and Management 44 1032–1048 (2008)

[44] Brian Bartell, Garrison Cotrell and Richard Belew “Automatic combination of multiple ranked retrieval systems” ACM SIGIR 1994

[45] Taeho Jo “Semantic Numerical Operations on Strings” IARIA, 2012

[46] Taher H. Haveliwala ” Efficient Encodings for Document Ranking Vectors” Technical Report 2002

[47] Olga Vechtomova, Murat Karamuftuoglu and Stephen E. Robertson On document relevance and lexical cohesion between query terms “Information Processing and Management 42 1230–1247 (2006)

[48] Tanveer Siddiqui and Umashanker Tiwary "A hybrid model to improve relevance in document retrieval" Journal of digital information management, Vol 4 Number 1, 2006

[49] F. Crestani "Application Of Spreading Activation Techniques In Information Retrieval" Artificial Intelligence Review Vol 11 (1997)

[50] Soni Dhanni "Wordnet: Database for English" ISSN 2231-1270 Volume 4, Number 1, pp. 35-39 (2012)

[51] The history of Information retrieval research: Mark Sanderson and W. Bruce croft.

[52] K. Spärck Jones, 'A statistical interpretation of term specificity and its application in retrieval', Journal of documentation, vol. 28, no. 1, pp. 11-21, 1972.

[53] K. Spärck Jones, Ed., Information Retrieval Experiment. Butterworth-Heinemann, 1981.

[54] G. Salton and C. S. Yang, 'On the Specification of Term Values in Autoatic Indexing', Department of Computer Science, Cornell University, Ithaca, New York, 14850, USA, TR 73-173, 1973

[55] G. Salton, A. Wong, and C. S. Yang, 'A vector space model for automatic indexing', Communications of the ACM, vol. 18, no. 11, pp. 613-620, 1975.

[56] Alfred V. Aho and Margaret J. Corasick, 'Efficient String Matching: An Aid to Bibliographic Search', Bell Laboratories Murray Hill, N.J. 07974. M. J. Corasick, The MITRE Corporation, Bedoford, Mass. O7130.

[57] Interpreting TF-IDF Term Weights as Making Relevance Decisions HO CHUNG WU and ROBERT WING PONG LUK

[58] M.W. Berry, S.T. Dumais & G.W. O'Brien 'Using Linear Algebra for Intelligent Information Retrieval'

[59] Juan Ramos 'Using TF-IDF to Determine Word Relevance in Document Queries' Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855

[60] H.P. Luhn, "A statistical approach to mechanized encoding and searching of literary information", IBM Journal

[61] Understanding TF-IDF [IRThoughts] "http://irthoughts.wordpress.com"

[62] Facebook warehousing 180 PETABYTES of data a year "http://www.theregister.co.uk/2012/11/09/facebook_open_sources_corona/"

[63] "Google process over 20 petabyte everyday" http://ebiquity.umbc.edu/blogger/2008/01/09/how-google-processes-20-petabytes-of-data-each-day/"

[64] The technique of spreading activation (Quillian, 1968; Cohen & Kjeldsen, 1987; Burke et al., 1997)

[65] WordNet: a lexical database for English by George Miller 1995 Princeton Univ., Princeton, NJ

[66] Larry Page, et al. “The PageRank citation ranking: bringing order to the web.”

[67] Rada Mihalcea "Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization" ACLdemo '04 Proceedings of the ACL 2004

[68] Aaron Bowen "Is Google Scholar a useful source?" Article CSU chico - Meriam Library

[69] Yongjing Lin, Wenyuan Li, Keke Chen, PhD and Ying Liu "A Document Clustering and Ranking System for Exploring MEDLINE Citations" J Am Med Inform Assoc. 2007 Sep-Oct; 14(5): 651–661

[70] Citation Analysis across disciplines: The impact of different data sources and citation metrics - “http://www.harzing.com/data_metrics_comparison.htm”

[71] Semi-Automatic Ontology Extension Using Spreading Activation by Wei Liu, Albert Weichselbraun, Arno Scharl, Elizabeth Chang

[72] Essential Science Indicators SM from Thomson Reuters – “http://archive.sciencewatch.com/about/met/”

[73] AMATI, G. AND VAN RIJSBERGEN, C. J. 1995. Probability, information and information retrieval. In Proceedings of the First International Workshop on Information Retrieval, Uncertainty and Logic (Glasgow, Sept.).

[74] Akiko Aizawa ”An information-theoretic perspective of tf–idf measures” PII: S 0 3 0 6 - 4 5 7 3 (0 2) 0 0 0 2 1 – 3 Parts of the results were presented at ACM SIGIR 2000

[75] Documentation for DBLP Bibliography Collection “http://sw.deri.org/~aharth/2004/07/dblp/dblp.html”

[76] S.E. Robertson, M.M. Hancock-Beaulieu “On the evaluation of IR systems” Volume 28, Issue 4, July–August 1992, Pages 457–466

[77] Gori, M., Pucci, A. “Research Paper Recommender Systems: A Random-Walk Based Approach” Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International

[78] Jomsri, P., Sanguansintukul, S.; Choochaiwattana, W. "A Framework for Tag-Based Research Paper Recommender System: An IR Approach" (WAINA), 2010 IEEE 24th International

[79] Kalervo Järvelin, Jaana Kekäläinen "IR evaluation methods for retrieving highly relevant documents" ACM SIGIR '00 conference on Research and development in information retrieval Pages 41-48

[80] Peng, Fuchun, and Andrew McCallum. "Accurate information extraction from research papers using conditional random fields." HLT-NAACL, 2004.

[81] Bo-Christer Björk, Annikki Roos and Mari Lauri "Scientific journal publishing: yearly volume and open access availability" IR information research vol. 14 no. 1, March, 2009

[82] Reyn Nakamoto, Shinsuke Nakajima, Jun Miyazaki, Shunsuke Uemura"Tag-Based Contextual Collaborative Filtering" IAENG International Journal of Computer Science, 34:2, IJCS_34_2_08

[83] Jöran Beel and Bela Gipp "Google Scholar's Ranking Algorithm: An Introductory Overview" International Conference on Scientometrics and Informetrics (ISSI'09), volume 1, pages 230–241

[84] Kim, Mary T. "Ranking of Journals in Library and Information Science: A Comparison of Perceptual and Citation-Based Measures" College and Research Libraries, v52 n1 p24-37 Jan 1991

[85] 1000 Most Common Words "http://www.giwersworld.org/computers/linux/common-words.phtml"

[86] Cohesion "http://en.wikipedia.org/wiki/Cohesion_(linguistics)"