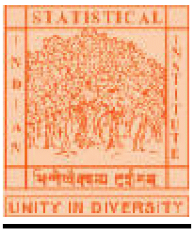
Using Negative Information for Text Retrieval

By

Aritra Banik

UNDER THE GUIDANCE OF
Dr. Mandar Mitra
Assistant Professor
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
OF
Master of Technology in **COMPUTER SCIENCE**



Indian Statistical Institute
203 Barrackpore Trunk Road,
Kolkata 700108

Date: 22nd July 2009

CERTIFICATE

I hereby forward the thesis prepared under my guidance by Aritra Banik, entitled **"Using Negative Information for Text Retrieval"** be accepted in partial fulfillment of the requirements for the Degree of Master of Technology in Computer Science of Indian Statistical Institute for the year 2009.

(Dr.Mandar Mitra)
Assistant Professor
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute

Acknowledgement

I would like to deeply thank the various people who, during the several months in which this endeavor lasted, provided me with useful and helpful assistance. Without their care and consideration, this thesis would likely not have matured.

First and foremost, we would like to thank **Dr. Mandar Mitra**, my supervisor and Project Guide, who introduced me to the world of the Information Retrieval and with his continuous support and encouragement helped me sail through it smoothly. It has really been a wonderful experience working in the stimulating environment of Information Retrieval lab under his guidance. I am really grateful to him for having faith in me and guiding me with his everlasting patience, invaluable insights, and timely feedbacks. The successful outcome of my work would have been impossible without him.

I would also like to thank **Mr. Sukomal Pal** who has helped me from the very first day I started working on my dissertation. He not only helped me understand the basics of Information Retrieval, but also helped me proceed to the advanced level of thoughts. He corrected me at every step and listened to all my ideas with utmost patience. He has also helped me prepare this report. I would like to heartily thank you Sukomalda for all your support. It would be no less than a sin, of not thanking **Dipasree Pal**, **Ayan Bandopadhyay** and **Samaresh Maiti** providing such a nice and friendly atmosphere in the lab and lending their helping hands whenever needed.

Dated: 22nd July 2009 **Aritra Banik**

CONTENTS:

CHAPTERS

Introduction

- Information Retrieval
- XML Retrieval
- Our Work
- References

1.1 Information retrieval

Information retrieval is a wide, often loosely-defined term unfortunately the word information can be very misleading. Nevertheless, 'information retrieval' has become accepted as a description of the kind of work published by Cleverdon, Salton, Sparck Jones, Lancaster and others. An information retrieval system does not inform (i.e. change the knowledge of) the user on the subject of his inquiry. It merely provides information on the existence (or non-existence) and whereabouts of documents relating to his request.

1.1.1 Definition

The discipline of information retrieval is almost as old as the computer itself. An old, if not the oldest, definition of information retrieval is the following by Mooers(1950)[2](recited from Savino and Sebastiani,1998[3]).
Information retrieval is the name of the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him.
In modern day terminology, an information retrieval system is a software program that stores and manages information on documents. The system assists users in finding the information they need.

1.1.2 How an IR system works

The information retrieval process consists of several steps. The basic steps of information retrieval are as follows

1.1.2.1 Stop word removal

Stop words are words with little meaning that are removed from the index and the query. Words might carry little meaning from a frequency (or information theoretic) point of view, or alternatively from a linguistic point of view. Words that occur in many of the documents in the collection carry little meaning from a frequency point of view. The words which have a grammatical function but do not convey any information about the subject matter of the document might be removed whether their frequency in the collection is high or low. In fact, they should especially be removed if their frequency is low, because these words affect document scores the most. Removing stop words for linguistic reasons can be done by using a stop list that enumerates all words with little meaning, like for instance the, it and a. Stop lists are used in many systems, but the lengths of the various stop lists may vary considerably. For instance, the Smart stop list

contains 571 words[5], whereas the Okapi system uses a moderate stop list of about 220 words (Robertson and Walker)[6].

1.1.2.2 Stemming

A stemmer applies morphological rules of humb to normalize words. The stemmers commonly used are those by Lovins[7] and Porter[8]. A stemmer can produce undesirable effects, for it may conflate two words with very different meanings to the same stem. For example 'operate', 'operating' and 'operations' are all stemmed to 'oper'. As a result, a query 'operating systems' can fetch documents related to 'operations research'.

1.1.2.3 Phrase extraction

During indexing and automatic query formulation, multiple words may be treated as one processing token. The meaning of a phrase might be quite different from what the two words independently suggest. A user who enters the query 'Stanford University' is less likely to be happy with a document which says 'Mr. Stanford never went to a university'. Maintaining the positional information of the terms can help to retrieve documents in which terms occur in the same relative positions as in the query. For example the query 'To be or not to be' is less likely to fetch Shakespeare's 'Hamlet' without positional information.

1.1.2.4 Index file structure

Within a document collection, we assume that each document has a unique number known as the document identifier (DocID). A weighted list of documents is constructed for every term in the collection, where the weight assigned to a document might be the number of occurrences of that term in it. The terms, which act as keys to their corresponding lists are kept sorted and are typically kept in memory whereas the associated lists (commonly referred to as *postings* are kept sorted by the list members' weights and are typically stored on secondary storage. For each query term, their postings are merged to give the final set of documents.

1.1.3 Evaluation of IR systems

Many different measures for evaluating the performance of information retrieval systems have been proposed. The measures require a collection of documents and queries. All common measures described here assume a ground truth notion of relevance: every

document is known to be either relevant or non-relevant with respect to a particular query. In practice, queries may be ill-posed and there may be various shades of relevance.

1.1.3.1 Evaluation of unranked retrieval sets

Given these ingredients, how is system effectiveness measured? The two most basic measures for information retrieval effectiveness are precision and recall. These are first defined for the simple case where an IR system returns a set of documents for a query. We will see later how to extend these notions to ranked retrieval situations.

Precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search, and **Recall** is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved)

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Fig. 1.1 shows the typical trend of decrease in precision with increased recall.

1.1.3.2 Evaluation of ranked retrieval sets

Another common measure used is the *Mean Average Precision* (MAP), which provides a single-figure measure of quality across recall levels. It is the average of the precision value obtained for the top set of k documents existing after each relevant document is retrieved, and this value is then averaged over information needs. That is, if the set of relevant documents for an information need $q_j \in Q$ is $\{d_1, \ldots d_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top result until you get to document d_k, then

$$\text{MAP}(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} \text{Precision}(R_{jk})$$

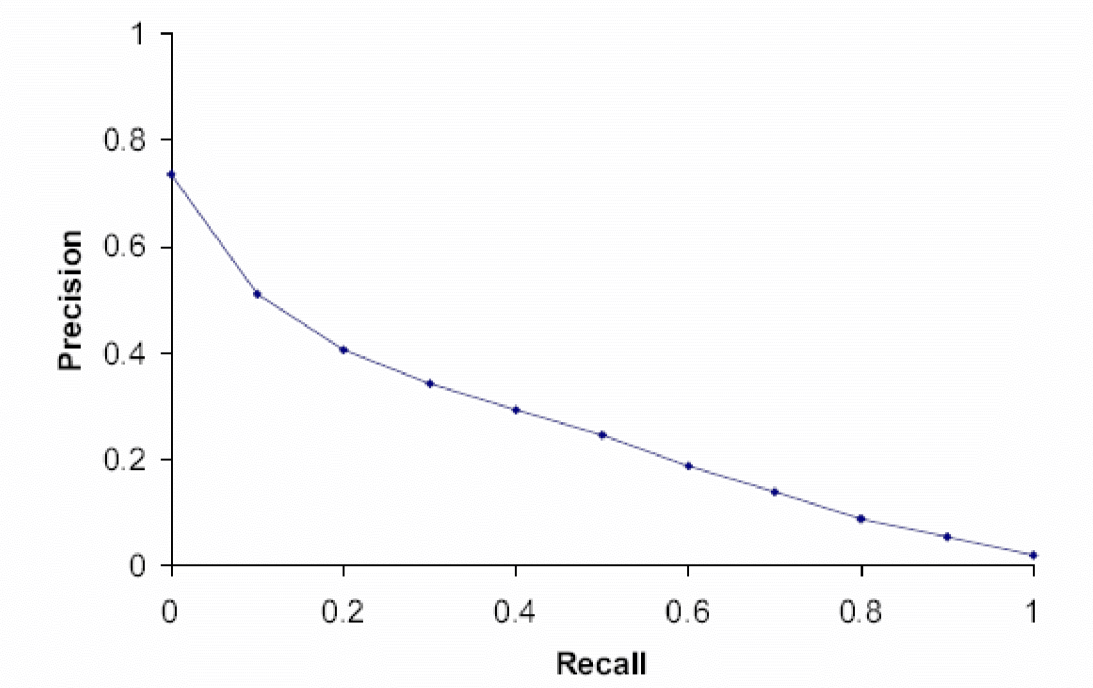


Figure1.1 A recall-precision graph

1.1.4 Information Retrieval Models

In the section, briefly discus two popular models used in IR: The Vector Space model and The Language modeling approach.

1.1.4.1 The Vector Space Model

The vector space model of information retrieval was developed by Salton and his students in the late 1960's and the early 1970 [4]. This model transforms any given text such as an article, a query, a portion of an article etc. into a vector in a very high-dimensional vector space. The main power of this model comes from its ability to measure the proximity between any two vectors, i.e., the 'closeness' between any two texts. In terms of information retrieval, when two vectors are close, then the corresponding texts are semantically related. The documents can then be ranked in decreasing order of their closeness to the query, yielding a semantic relatedness ranking, as desired in modern information retrieval systems. Salton and his students also implemented a system based on the vector space model, the Smart system [5]. Through Smart, the vector space model has had a tremendous influence on IR.

Relevance rankings of documents in a keyword search can be calculated, using the assumptions of document similarities theory, by comparing the deviation of angles between each document vector and the original query vector where the query is represented as same kind of vector as the documents. In practice, it is easier to calculate the cosine of the angle between the vectors instead of the angle:

$$\cos\theta = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \, \|\mathbf{v_2}\|}$$

A cosine value of zero means that the query and document vector are orthogonal and have no match (i.e. the query term do not exist in the document being considered).

tf-idf weights

In the classic vector space model proposed by Salton, Wong and Yang the term specific weights in the document vectors are products of local and global parameters. The model is known as term frequency-inverse document frequency model. The weight vector for document *d* is $\mathbf{v}_d = [w_{1,d}, w_{2,d}, \ldots, w_{N,d}]^T$, where

$$w_{t,d} = \mathrm{tf}_t \cdot \log \frac{|D|}{|\{t \in d\}|}$$

and

tf_t is term frequency of term *t* in document *d* (a local parameter)

$\log \frac{|D|}{|\{t \in d\}|}$ is inverse document frequency (a global parameter).

$|D|$ is the total number of documents in the document set;

$|\{t \in d\}|$ is the number of documents containing the term *t*.

1.1.4.2 Language Models

Basic language modeling ideas have been used in information retrieval and document classification for quite some time. In the so-called *naive Bayes* text classification method, a unigram language model is estimated for each class, and then combined with a class prior to form the posteriors used for classification; the naivety of the approach lies in the unrealistic independence assumptions that lead to a unigram model. While the independence assumptions are clearly incorrect, such "bag of words" models are surprisingly effective for classifying documents according to a small number of predefined labels.

A similar approach is adopted in the standard probabilistic model of document retrieval first proposed by Robertson and Sparck Jones [9] [10] [11]. In this model, distributions over documents are estimated for two classes: "relevant" and "non-relevant." Documents are broken down into attributes, in the simplest case indicating occurrence or non-occurrence of

individual words, and the attributes are modeled independently, as in the naive Bayes model for classification. In contrast with document classification, however, for retrieval there is typically little, if any, training data, and the only evidence available for estimating the models is the query itself. Thus, one is led to model the distribution of the query terms in relevant and non-relevant documents. The Okapi system [6] has been one of the primary vehicles for the Robertson-Sparck Jones model of retrieval, and has met with considerable empirical success.
The fact that so little evidence is available for estimating the relevant and non-relevant document classes has made it attractive to consider "turning the problem around." In 1998 Ponte and Croft [12] proposed using a smoothed version of the document unigram model to assign a score to a query, which can be thought of as the probability that the query was generated from the document model. This simple approach was remarkably effective "right out of the box." As developed further in [13], this approach can be thought of as using a language model as a kind of noisy channel model or "translation model" that maps documents to queries.

1.2 XML Retrieval

1.2.1 Definition

XML Retrieval, or XML Information Retrieval, is the content-based retrieval of documents structured with XML (eXtensible Markup Language).[14]
The fundamental difference between standard information retrieval and XML retrieval is the unit of retrieval. In traditional IR, the unit of retrieval is fixed: it is the complete document. In XML retrieval, every XML element in a document is a retrievable unit. This makes XML retrieval more difficult: besides being relevant, a retrieved unit should be neither too large nor too small.

1.2.2 Evaluation of XML retrieval system

The premier venue for research on XML retrieval is the INEX (**IN**itiative for the **E**valuation of **X**ML retrieval) program, a collaborative effort that has produced reference collections, sets of queries, and relevance judgments. A yearly INEX meeting is held to present and discuss research results. The INEX 2002 collection consisted of about 12,000 articles from IEEE journals. The IEEE journal collection was expanded in 2005. Since 2006 INEX uses the much larger English Wikipedia as a test collection. The relevance of documents is judged by human assessors.

Two types of information needs or topics in INEX are content-only or CO topics and content-and-structure (CAS) topics. *CO topics* are regular keyword queries as in unstructured information retrieval. *CAS topics* have structural constraints in addition to keywords.
Here we present two famous evaluation metric Interpolated Precision and MAiP.

1.2.2.1 Interpolated Precision

Interpolated precision combines both recall and precision. Let *r* denote a certain recall level (e.g., 30%). *k*(*r*) is used to denote the first value of *k* so that *recall@k* >= *r*. The interpolated precision at a recall level *r* is defned to be *max{precision@k* : *k* >= *k*(*r*)}. The interpolated precision measures the maximal precision that can be achieved above a recall level threshold.

1.2.2.2 AiP

In addition to using the interpolated precision measure at selected recall levels, we also calculate overall performance scores based on the measure of average interpolated precision AiP. For an INEX topic, we calculate AiP by averaging the interpolated precision scores calculated at 101 standard recall levels (0.00, 0.01, . . . , 1.00).

$$AiP = \frac{1}{101} \cdot \sum_{x=0.00,0.01,\ldots,1.00} iP[x]$$

1.2.2.3 MAiP

Performance across a set of topics is measured by calculating the mean of the AiP values obtained by the measure for each individual topic, resulting in mean average interpolate precision (MAiP). Assuming there are n topics

$$MAiP = \frac{1}{n} \cdot \sum_{t} AiP(t)$$

1.3 Our Work

1.3.1 Motivation

Present-day search engines and most of the static IR systems expect a few terms or a phrase as information need, whereas a common user is still

comfortable using a natural language query. His information need is therefore often not properly reflected in the query. The system also sometimes fails miserably in query understanding and thereby meeting user's need. Our initial experiments using standard query sets over semi-structured (INEX Wikipedia corpus) data show that some queries contain negative information (i.e. information about what the user is not looking for) which IR models such as the Vector Space Model cannot distinguish from the positive information and eventually lead to poorer retrieval performance in terms of precision and recall.

1.3.2 Problem Statement

Though IR can be pedagogically argued as a natural subfield of NLP as it deals with a particular application of natural language processing, in actuality, their interactions have been very limited. But with the increase of quantitative methods in NLP, some of the newer demands of IR can be met by application of NLP [15]. One such area in IR is query processing. As Lewis and Karen Spärck Jones observed "many end users have little skill or limited experience in formulating initial search requests and modifying their requests after observing failure. Even while relevance feedback is available, it needs to be leveraged from a sensible starting point" [16]. The problem is much worse in the field of semi-structured retrieval. While formulating structure-conscious queries, we found that as high as 63% queries were detected to contain syntactic errors [17]. Notable point is that these queries were prepared by experienced IR researchers. A typical query - usually a small number of keywords, with or without additional structural constraints - is frequently an inadequate representation of a user's information need. Had the user been allowed to pose a query in natural language, he would have freedom to verbosely express his information need: what he precisely wants and what he does not. One specific drawback of keyword-based topics is the inability to clearly describe what the user is NOT looking for. As part of an initial study, we looked at INEX queries 414-543 for semi-structured text retrieval over INEX wikipedia English collection. For a reasonably large number of queries, we observed that along with their information needs, users also explicitly express what they do not want or what documents or texts are not relevant to them. This information usually occurs only in the NARRATIVE part of the topic which is not generally used during retrieval. We also find that there are three types of keywords in the entire topic: (i) POSITIVE terms which the user stresses on and whose presence is mandatory to qualify a text to be relevant and the (ii) NEGATIVE terms whose presence will basically disqualify the text to be relevant and (iii) a set of WEAK terms whose presence does not matter much. Indexing positive terms and weak terms with the same weighting scheme dilutes the importance of a really relevant document and sometimes puts it on the same plane as a non-relevant one. The situation becomes worse when

negative terms also receive the same weightage with that of positive and weak terms. A completely irrelevant document may get a good retrieval score and achieve high rank. Both precision and recall suffer when non-relevant documents achieve high rank and relevant documents are not retrieved because of topic dilution or topic drift caused by the presence of negative and weak terms. For example, INEX query 439 shown in figure 1.2. For this query it seems that keyword based retrieval will include results containing "**album whose lyrics include Japanese and Jews**" although it is irrelevant.

I am going to write a history class report on the interactions between Jews and Japanese during and around the Second World War.A relevant element must describe interactions or connections between Jews and Japanese. An element that describes both Jews and Japanese separately without making any connections is irrelevant. An element is irrelevant if its topic is insignificant. For example, if there is a **minor pop music album whose lyrics include Japanese and Jews during the Second World War, the element is irrelevant.** An element is also irrelevant, if from the element alone it's unclear when the interactions happened.

Figure 1.2 INEX query 439

1.3.3 Experimental Frame work

Retrieval Engine	SMART
Programming Language	C
Query and Document Set	INEX2007 and INEX2008

1.4 References

1. WINOGRAD, T., *Understanding Natural Language,* Edinburgh University Press, Edinburgh (1972).
2. Mooers, "Information Retrieval viewed as temporal signalling," 1950, Proceedings of the International Congress of Mathematicians.
3. Savino P. and S. Sebastiani, "Essential bibliography on multimedia information retrieval, categorization and filtering," 1998, In slides of the 2nd Europen Digital Libraries Conference Tutorial on Multimedia Information Retrieval
4. C.S. Yang Gerard Salton, A. Wong, "A vector space model for Information Retrieval," .
5. Smart, "ftp://ftp.cs.cornell.edu/pub/smart,"

6. S.E. Robertson and S.Walker, “Okapi, Keenbow at TREC-8,” 2000, Eigth Text Retrieval Conference
7. Lovins, “Development of a stemming algorithm,” Mechanical translations and Computational Linguistics,vol. 11, pp. 22–31, 1993.
8. M.F. Porter, “An algorithm for suffix stripping,” Program 14, pp. 130–137, 1980.
9. S. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27, 129-146, 1976.
10. K. Sparck Jones, S. Walker and S. Robertson, A probabilistic model of information retrieval: development and comparative experiments, (Part 1). *Information Processing and Management*, 36, pp. 779-808, 2000.
11. The probabilistic retrieval model, http://web.soi.city.ac.uk/research/cisr/okapi/prm.html.
12. J. Ponte and W. B. Croft. A language modeling approach to information retrieval. *Proceedings of the ACM SIGIR*, pp. 275-281, 1998.
13. A. Berger and J. Lafferty, Information retrieval as statistical translation, in Proceedings of the *1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222-229, 1999.
14. XML-Retrieval, Wikipedia, the free encyclopedia
15. C. Manning and H. Schutze. Foundations of Statistical Natural Language Process-ing. MIT Press, Cambridge, MA, 1999
16. D. D. Lewis and K. S. Jones. Natural language processing for information retrieval.
17. Commun. ACM, 39(1):92–101, 1996.B. Sigurbj¨ornsson and A. Trotman. Queries: INEX 2003 working group report.In Proceedings of the 2nd workshop of the initiative for the evaluation of XML retrieval (INEX), pages 175–178, 2003.

Our Work

- Introduction
- Manual detection of negation present in XML queries
- Attempt with only positive sentences
- Implementation of the feedback model
- Experiment to modify term frequency
- References

1.5 Introduction

XML topics are more specific than their traditional IR counterparts. A user categorically specifies what (s) he wants and more importantly what (s) he does not want. Consider the example of query 426 shown in figure 2.1.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE inex_topic SYSTEM "topic.dtd">
<inex_topic topic_id="426"  ct_no="30">
<title>Ajax Asynchronous JavaScript and XML programming technologies
applications</title>
<castitle>//article[about(.,Ajax Asynchronous JavaScript and XML
programming technologies applications)]</castitle>

<description>I want information about AJAX (Asynchronous JavaScript
and XML) programming : which technologies are used, which kind of
applications are based on Ajax.</description>
<narrative>I am a webmaster, I need to update my knowledge about new
web technologies.I'd like to know what is Ajax programmming. Which
technologies are used by Ajax development ? Which kind of
applications can we imagine for dynamic web sites ? I'd like to have
an idea on how Ajax might change the Web.I don't need information
about Ajax libraries or Javascript engines.
</narrative>
</inex_topic>
```

Figure 2.1 INEX 2007 query, Query 426

If we look at the narrative part of the query, we observe that the lines shown in bold are describing what the user actually wants while the other line describes what the user does not want. Search systems based on a bag of words approach typically do not distinguish between keywords that a user actually wants and keywords that the user specifically does not want. As a result, such systems may retrieve documents which should not be retrieved. For example, for the query in figure 2.1 traditional search engines will usually retrieve pages which contain "Ajax libraries" and "Javascript engines" as well. Thus mentioning what the user does not want may actually degrade the performance of the search engine. Throughout this discussion, we describe sentences which depict what user does not want as *negative sentences* and other sentences as *positive sentences*. Although detection of negative sentences seems easy from the above example, this may not be the case for all queries. Let us take the example of query 431 shown in figure 2.2.

```
<narrative>I am supposed to use OpenGL in my computer graphics
course. An element speaking about OpenGL in computer graphics
should be considered as relevant whereas an element speaking
about computer graphics without mentioning OpenGL should be
considered as irrelevant.</narrative>
```

Figure 2.2 INEX 2007 query, Query 431

In this query, the clause "an element speaking about computer graphics without mentioning OpenGL should be considered as irrelevant" appears to be the negative part, but here only "computer graphics" is negative whereas "OpenGL" is a positive word. Our work can be divided into two parts: (i) Detection of negative information present in XML queries, and (ii) Use of these negative and positive sentences for improved retrieval. Our focus is mainly on the second part, the first task is nontrivial natural language processing kind of problem is our secondary goal. Currently, we manually label sentences as positive or negative, and explore ways to use those labels to improve retrieval effectiveness.

1.6 Manual detection of negation present in XML queries

In manual detection phase, we read the narrative section of all queries and the sentences carrying a positive sense are separated from the sentences carrying a negative sense. File containing positive sentences for query no 426 (which is shown in figure2.1) is shown in figure 2.3.

```
<title>Ajax Asynchronous JavaScript and XML programming
technologies applications</title>
<castitle>//article[about(.,Ajax Asynchronous JavaScript and XML
programming technologies applications)]</castitle>
<description>I want information about AJAX (Asynchronous
JavaScript and XML) programming : which technologies are used,
which kind of applications are based on Ajax.</description>
<narrative>I am a webmaster, I need to update my knowledge about
new web technologies.I'd like to know what is Ajax programmming.
Which technologies are used by Ajax development ? Which kind of
applications can we imagine for dynamic web sites ? I'd like to
have an idea on how Ajax might change the Web.
</narrative>
```

Figure 2.3 Positive file for query no 426

It may happen that a query may not contain any negative part. In that case, narrative part is empty in the negative file.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE inex_topic SYSTEM "topic.dtd">
<inex_topic topic_id="426"  ct_no="30">
<title>Ajax Asynchronous JavaScript and XML programming technologies
applications</title>
<castitle>//article[about(.,Ajax Asynchronous JavaScript and XML
programming technologies applications)]</castitle>

<description>I want information about AJAX (Asynchronous JavaScript
and XML) programming : which technologies are used, which kind of
applications are based on Ajax.</description>
<narrative>I don't need information about Ajax libraries or
Javascript engines.
</narrative>
</inex_topic>
```

Figure 2.3 Negative file for query 426

1.7 Attempt with only positive sentences

1.7.1 Approach

We start our experiment by considering only the positive sentences. We first create positive and negative files for the INEX2007 topics. The idea is to run the 130 positive queries and compare them with the original run. One point to note here is that although we are running the system on 130 queries only 68 out of these 130 contain some negative part .The remaining queries do not contain any negative information.

1.7.2 Results

Table 2.2 and table 2.3 compare the results obtained using the original query and the positive query. Only those queries for which performance is affected by 5% or more are listed. Negative value implies degradation in the result and a positive value implies improvement in the result. Performance change with respect to the original run for INEX 2007 and INEX 2008 queries are shown in table 2.1.

	Total no of queries	No of queries where result is improved	No of queries where result is degraded
INEX 2007	68	22	10
INEX2008	82	17	4

Table 2.1 Performance change with respect to the original run for INEX 2007 and INEX 2008 queries

Query No	Difference in similarity score	Percentage improvement	Query No	Difference in similarity score	Percentage improvement
427	0.0113	5.98	465	0.0273	32.23
426	0.0188	6.23	483	0.0142	35.59
475	0.0158	6.33	490	0.0622	36.76
421	0.0152	7.31	439	0.0444	38.88
508	0.0125	9.16	473	0.2177	56.81
484	0.0098	9.41	471	0.2593	98.63
535	0.0442	9.54	467	-0.1331	-49.35
500	0.0388	9.96	481	-0.0125	-15.78
480	0.0195	10.1	489	-0.0087	-14.01
538	0.0201	10.24	497	-0.0184	-13.56
417	0.0241	10.39	485	-0.0341	-12.4
542	0.0246	13.44	505	-0.0185	-10.93
482	0.0441	14.55	491	-0.0087	-8.91
470	0.0759	16.21	416	-0.0138	-7.39
520	0.0378	21.67	463	-0.0319	-6.99
507	0.03	23.2	469	-0.0148	-6.28

Table 2.2 Comparative study between retrieval with original query and retrieval with positive query for INEX2007

Query No	Difference in similarity score	Percentage improvement	Query No	Difference in similarity score	Percentage improvement
576	0.0073	5.06	677	0.026	30.09
626	0.0202	5.47	556	0.0408	30.58
545	0.0221	5.74	597	0.1089	40.57
641	0.0146	6.64	544	0.022	46.61
613	0.0253	6.76	644	0.1015	58.03
595	0.0198	7.13	673	0.0957	59.37
553	0.0201	9.02	551	-0.0319	-11.98
610	0.0354	9.46	616	-0.0293	-7.33
642	0.033	15.46	561	-0.017	-6.11
659	0.0142	22.68	675	-0.0146	-5.34
574	0.0461	25.81			

Table 2.3 Comparative study between retrieval with original query and retrieval with positive query for INEX2007

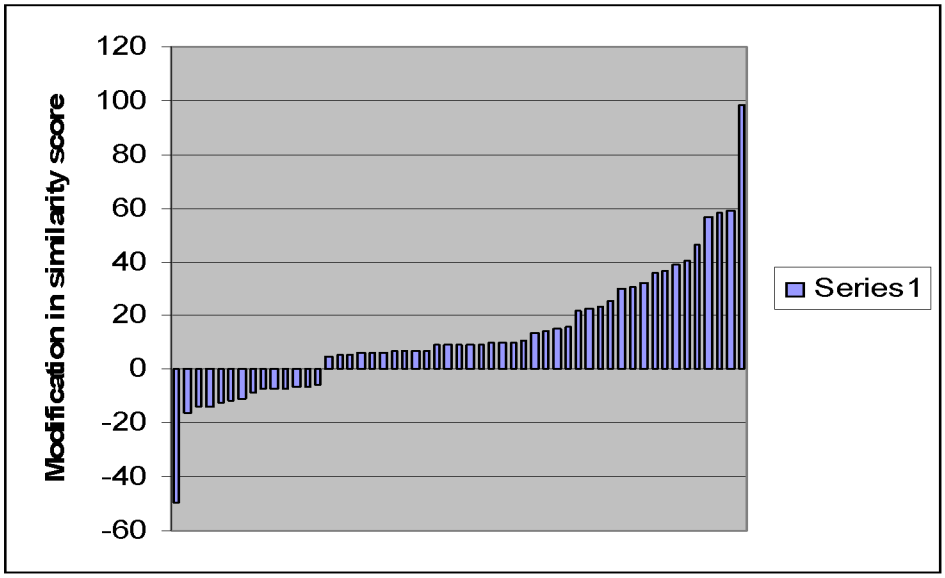


Figure2.4 Comparative study between retrieval with original query and retrieval with positive query for INEX2007 and INEX2008 using bar chart

1.7.3 Analysis

The result of both INEX 2007 and INEX 2008 are quit promising. On examining the result more carefully, we find that a number of queries contain negative sentences which are of little use from the perspective of keyword based information retrieval. Take the example of query no 420, where the user needs information about Phong Shading but the user is not interested in documents about shading models. The actual negative sentence is "*Information about shading models alone, without any discussion of Phong Shading model is not relevant*". The requirements of the user are clear from the narrative. However simply deleting the sentence also results in the elimination of important term such as "*shading model*" and "*Phong shading*", Thus, using the positive query does not improve performance for the query. Another example is query no 534 which is shown in figure 2.5. The negative part of the query contains only the following two keywords: *paintings, painters*. These keywords are not strongly indicative of non-relevance. Thus, the removal of such sentences is of little help.

```
<narrative>As a student I am invited to do a website on Vincent
Van Gogh, and one part should deal with its paintings. I would
like to introduce some images to illustrate that. Each painting
from Van Gogh is relevant. Paintings from other painters are not
relevant.</narrative>
```

Figure2.5 Narrative part of query no 534

1.8 Implementation of the feedback model

1.8.1 Approach

Now we try to use negative sentences for improving retrieval performance. This is one of the most difficult part of the project. The difficulty hidden inside our basic retrieval strategy. Like most of the retrieval systems, in SMART you can guide the search engine that what to search but the difficulty is to tell the search engine not to retrieve something. First we try to implement it with the help of the feedback model present in the SMART system. The idea is to feed the negation in the feedback model present in SMART system but we found that it is not a good idea because SMART maintains its own data structures and it's very difficult to change those data structure in runtime. Our strategy is to run both positive and negative queries and then modify the retrieval result obtained using the positive query on the basis of results obtained using the negative query. Thus if a document is retrieved by both queries, and the similarity score in the two runs are respectively s1 and s2, the final score assigned to the document is some function of s1 and s2.

1.8.2 Algorithm

The algorithm for this strategy is shown in figure 2.8. The function f can be any function such that f(a,b)<f(a) for all a, b. We have used f (a, b) =a-b. Point to be noted here is we have used the title, description, narrative

```
FOR each query q
    FOR each document d retrieved for query constructed
    by positive part of query q with a similarity score
    Score_positive
        IF d is also retrieved for query constructed by
        negative part of query q with a similarity score
        score_negative
            score_new=f(score_positive, score_negative)
        ELSE
            score_new=score_positive
        END IF
    END FOR
END FOR
```

Figure 2.8 Algorithm to implement feedback model

fields for the positive query and the narrative only for negative queries because it is observed that the title and description does not contain any negative part.

1.8.3 Results

Query No	Difference in similarity score	Percentage improvement	Query No	Difference in similarity score	Percentage improvement
427	0.0118	6.24	483	0.0143	35.84
426	0.0189	6.26	490	0.0622	36.76
421	0.0152	7.31	439	0.0449	39.32
538	0.0173	8.82	473	0.2189	57.12
508	0.0125	9.16	471	0.2593	98.63
484	0.0099	9.51	416	-0.1455	-77.89
535	0.0452	9.76	497	-0.0682	-50.26
480	0.0195	10.1	467	-0.133	-49.31
417	0.0241	10.39	481	-0.0125	-15.78
431	0.0476	11.64	489	-0.0086	-13.85
542	0.0246	13.44	485	-0.0341	-12.4
482	0.0441	14.55	505	-0.0185	-10.93
470	0.0759	16.21	463	-0.0496	-10.87
520	0.0378	21.67	491	-0.0087	-8.91
507	0.03	23.2	469	-0.0148	-6.28
465	0.02	23.61	498	-0.0041	-5.2

Table2.4 Comparative study between retrieval with original query and retrieval with feedback strategy for INEX2007

Query No	Difference in similarity score	Percentage improvement	Query No	Difference in similarity score	Percentage improvement
576	0.0073	5.06	677	0.026	30.09
626	0.0204	5.52	556	0.0408	30.58
545	0.0221	5.74	597	0.1012	37.7
595	0.02	7.2	544	0.022	46.61
553	0.0201	9.02	644	0.1008	57.63
610	0.0354	9.46	551	-0.0319	-11.98
642	0.033	15.46	613	-0.0351	-9.37
641	0.0398	18.09	616	-0.0291	-7.28
659	0.014	22.36	550	-0.0335	-6.65
574	0.0455	25.48	561	-0.017	-6.11
673	0.0431	26.74	675	-0.0146	-5.34

Table2.5 Comparative study between retrieval with original query and retrieval with feedback strategy for INEX2008

Performance change with respect to the original run for INEX 2007 and INEX 2008 queries are shown in table 2.6. If we compare table 2.1 and table 2.6 we can clearly see the approach yields inferior results for both datasets.

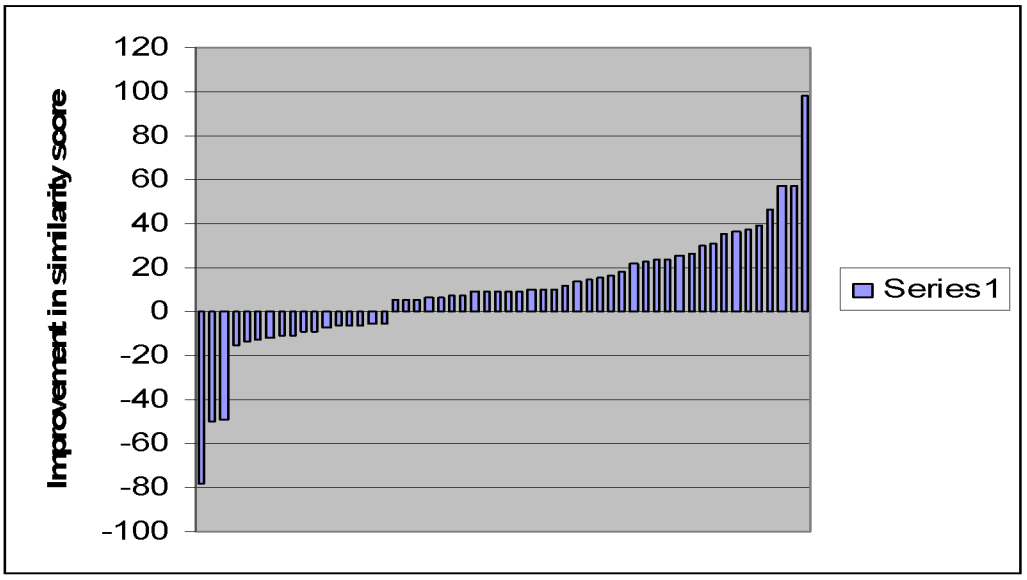


Figure2.9 Comparative study between retrieval with original query and retrieval with feedback strategy for INEX 2007 and INEX2008 using bar chart

	Total no of queries	No of queries where result is improved	No of queries where result is degraded
INEX 2007	68	21	11
INEX2008	82	16	6

Table 2.6 Performance change with respect to the original run for INEX 2007 and INEX 2008 queries

1.8.4 Analysis

We start our analysis with two queries for which the algorithm is giving worse results than the result present in table 2.1, query no 416 and query no 498. Query no 416 is shown in figure 2.10. Main keywords in the negative query are *woodwind chamber music and instrument*. The negative query retrieves all documents mainly about woodwind chamber

music and its instruments. The positive query is also asks for the same thing. So the negative part which supposes to retrieve some thing the user does not want, actually retrieving some thing that user wants. As a result our strategy is showing poor result. Similar thing happens for query 498. Thus, simply reducing the similarity score does not seem like a good idea.

```
<narrative>The relevant documents should talk about
woodwind chamber music, either what is composition of the different
ensembles, the name of the composers who wrote for woodwind chamber
ensembles and works for such ensembles. A document that simply quote
or describe an instrument which is used for woodwind chamber music is
not relevant.A document that simply gives the name of a given
ensemble is not relevant.</narrative>
```

Figure 2.10 Query no 416

In further analysis we found out that in most of the cases, positive and negative sentences are so close to each other that in many of the documents they occur simultaneously. Here the problem is it is clear from the query that what user wants and what not; but it is not clear that if both of them occur whether they want that document or not. We assumed that if a single element is relevant then the whole document is relevant. In many of the cases where result is degraded, similarity score of some document reduces as the document contains both positive and negative terms. For example, let us look at query 467 shown in figure 2.6. Here the user is interested in text mining but not interested in data mining. It is not clear from the query whether a document containing both topics is relevant or not. During our analysis we found

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE inex_topic SYSTEM "topic.dtd">
<inex_topic topic_id="467"  ct_no="91">
<title>Definition and ressources on text mining</title>
<castitle>//*[about(., "Definition and ressources on text
mining")]</castitle>
<description>I would like to know what is text mining</description>
<narrative>I would like to know what is text mining and to find
ressources : links, tools, corpora.
I am also interested in biomedical text mining.
I am not interested by data mining or mining.</narrative>
</inex_topic>
```

Figure 2.6 Query no 467

that data mining is relevant or not. During our analysis we found that data mining and text mining are so closely related that in many of the documents they occur simultaneously. For some documents marked

relevant, we observe that the focus of the document is data mining, with only one or two lines addressing text mining.
Again consider query 481 shown in figure 2.7. A user wants information about Asian news channels but does not need information about worldwide news channels. When we analyzed the results we found that many relevant documents contain a list of worldwide news channels including Asian news channels as well. As these documents contain information about Asian news channels they are surely relevant but due to deletion of negative sentences the phase" *worldwide news channels*" is deleted. As a result of such deletion similarity score between these documents and the query reduces which in turn affects our result.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE inex_topic SYSTEM "topic.dtd">
<inex_topic topic_id="481"  ct_no="116">
<title>asia "news channel"</title>
<castitle>//article[about(., "news channel" + asia)]</castitle>
<description>Find articles about any of the Asian news
channel.</description>
<narrative>The TV channels which are dedicated for News alone are
gaining enormous popularity. The query is aimed at finding news
channels which are from asian countries. For a document to be
relevant, it should include the name of the news channel along with
an asian country name.If it includes more information about the news
channel it will be considered more relevant.Worldwide News channels
like BBC and CNN are considered as irrelevant to the
query.</narrative>
</inex_topic>
```

Figure 2.7 Query no 481

1.9 Experiment to modify term frequency

1.9.1 Approach

All the analysis suggests that one of the reasons for degradation of result is the reduction in term frequency for some of the positive terms. To address this problem we propose two algorithms. We first try duplicating the narrative part in the positive file if the original file contains some negative part. This increases the term frequency of all terms in the positive part of the query. Next we attempt to selectively increase the term frequency of positive terms whose term frequency is reduced due to deletion of negative sentences. Same time we delete terms from the positive part of the query if their term frequency is very high in the negative part of the query as such terms are likely to be negative terms.

1.9.2 First algorithm

```
FOR all query q
        IF negative part not equals to null
                double the narrative of positive file
        END IF
END FOR
```

Figure2.11 First algorithm to modify term frequency

1.9.3 Results

Query No	Difference in similarity score	Percentage improvement	Query No	Difference in similarity score	Percentage improvement
538	0.014	7.14	491	-0.0172	-17.62
500	0.0405	10.4	485	-0.0478	-17.38
417	0.0242	10.43	497	-0.0218	-16.06
520	0.0248	14.22	433	-0.0443	-15.65
521	0.033	17.51	441	-0.0271	-14.48
490	0.0315	18.62	498	-0.0114	-14.47
542	0.0362	19.77	454	-0.044	-13.97
439	0.0298	26.09	505	-0.023	-13.59
471	0.168	63.9	458	-0.0242	-9.06
473	0.2655	69.28	448	-0.023	-8.54
467	-0.1338	-49.61	463	-0.033	-7.23
481	-0.0342	-43.18	476	-0.0369	-7.01
507	-0.0512	-39.6	421	-0.0123	-5.91
480	-0.0733	-37.96	434	-0.0172	-5.85
541	-0.0336	-24.85			

Table 2.7 Result of first algorithm for INEX2007 queries

	Total no of queries	No of queries where result is improved	No of queries where result is degraded
INEX 2007	68	11	20

Table 2.8 Performance change with respect to the original run for INEX 2007

1.9.4 Analysis

The results of this algorithm are poor but this is not very unexpected. We are blindly doubling the narrative of positive queries, which is not a very good way to modify term frequency.

1.9.5 Second Algorithm

As mentioned earlier in the second approach we try to identify a term as positive or negative term depending on its term frequency in the positive and negative parts of the query.

```
FOR each term w present in the document
        let wp be the term frequency of w in the positive
        query and wn be the term frequency of w in the
        negative query

        IF wn>0 and wp>0
                IF wp-wn>positive_threshold
                        add wn no of w's to the positive query
                END IF
                IF wn-wp>negative_threshold
                        add wp no of w's to the negative query
                END IF
        END IF
END FOR
```

Figure 2.12 Second Algorithm

1.9.6 Results

The result of this strategy is shown in figures 2.18, 2.19 and 2.20. Here *positive_threshold* and *negative_threshold* are two parameters. Appropriate settings for these parameters were found by trial and error.

	Total no of queries	No of queries where result is improved	No of queries where result is degraded
INEX 2007	68	22	8
INEX2008	82	19	5

Table 2.9 Performance change with respect to the original run for INEX 2007 and INEX 2008 queries

Query No	Difference in similarity score	Percentage improvement	Query No	Difference in similarity score	Percentage improvement
475	0.0131	5.24	465	0.0187	22.08
427	0.0101	5.34	507	0.03	23.2
426	0.0165	5.47	508	0.0449	32.89
421	0.0142	6.83	439	0.0401	35.11
499	0.0361	9.02	483	0.0142	35.59
500	0.0353	9.06	490	0.076	44.92
535	0.0428	9.24	473	0.2179	56.86
480	0.0195	10.1	467	-0.1331	-49.35
538	0.0201	10.24	489	-0.0099	-15.94
417	0.0241	10.39	481	-0.0118	-14.9
542	0.0246	13.44	497	-0.0184	-13.56
470	0.0759	16.21	505	-0.0219	-12.94
520	0.0291	16.69	491	-0.0063	-6.45
482	0.0606	20	469	-0.0148	-6.28
471	0.0548	20.84	463	-0.0248	-5.43

Table 2.10 Comparative study between retrieval with original query and retrieval with modified positive query with term frequency for INEX2007

Query No	Difference in similarity score	Percentage improvement	Query No	Difference in similarity score	Percentage improvement
600	0.0092	4.88	677	0.017	19.68
576	0.0073	5.06	659	0.0142	22.68
647	0.0122	5.64	574	0.0432	24.19
559	0.0175	6.29	544	0.0124	26.27
641	0.0146	6.64	556	0.038	28.49
613	0.0267	7.13	673	0.0547	33.93
595	0.0252	9.07	644	0.1015	58.03
610	0.0345	9.22	551	-0.0319	-11.98
546	0.0062	9.38	669	-0.0342	-9.28
597	0.0312	11.62	616	-0.0293	-7.33
553	0.0271	12.16	561	-0.017	-6.11
642	0.0297	13.92	675	-0.0145	-5.31

Table 2.11 Comparative study between retrieval with original query and retrieval with modified positive query with term frequency for INEX2008

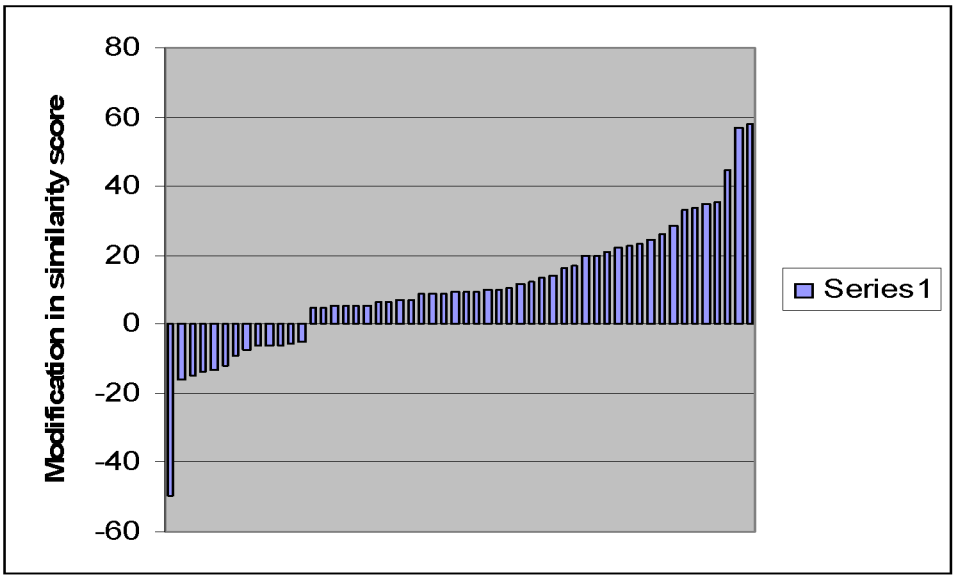


Figure2.13 Comparative study between retrieval with original query and retrieval with modified positive query with term frequency for INEX2007 and INEX 2008 using bar chart

1.9.7 Analysis

Although our experimental result implies that in case of 8 queries the quality of search result is degraded by the second algorithm when we manually checked the results, we found some irrelevant documents marked as relevant. This is the reason behind the poor results obtained for some queries. For example in case of query 505, a document which is about industrial applicability is marked as relevant, whereas query 505 asks for documents about software patents. Similarly in case of query 481, a document which is about Rizwan Khan, a BBC reporter is marked as relevant, whereas query 481 asks for documents about Asian

```
Find the plants and trees that are known to cause allergy problems
(hay fever).To be relevant document (or part of a document) should
cite a species.If only indicating that some pollen may cause allergy,
it is not relevant.Treatments and medications are not relevant.
```

Figure 2.14 Query No 551

news channels. Similar anomalies were observed for some other INEX 2007, INEX 2008 queries for which our approach is showing poor result. From the INEX 2008 query set query no 551(shown in figure 2.21) asks for documents about pollen allergy. The query clearly mentions that treatments and medications are irrelevant, but a document about Meniere's disease and a document about Cetirizine which is used for allergic treatment is marked as relevant.

1.10 Automatic Negation Detection

1.10.1 Approach

We used a fairly simple algorithm to find out negation present in the query. As the queries are coming from different sources and from different users it's very difficult to train the system. Out of 130 INEX2007 queries, we have chosen the first 65 as our training set and the rest as the test set. In the first step we have extracted all adjectives and verbs from the query collection and measured the term frequency of each phrase in positive sentences and negative sentences. Depending on these two scores we have assigned a composite score to each of the terms. The reason we have chosen only adjective and verb is that such words (irrelevant, don't want) mainly determines whether a sentence is positive or negative. Depending on this score we have selected *n* most negative phrases. We have done negation detection on the rest of the queries. If these terms occur at odd number of times in a sentence we mark that sentence as negative (we assumed that an even number of occurrences of negative words makes a sentence positive). For example, consider query 414 shown in figure 2.21. The sentence shown in bold is a positive sentence where two negative

```
<narrative>To solve an argument with a friend about hip hop
music and beats, I want to learn all there is to know about hip
hop beats. I want to know what is meant by hip hop beats, what
is considered a hip hop beat, what distinguishes a hip hop beat
from other beats, when it was introduced and by whom. I
consider elements relevant if they specifically mention beats
or rythm. Any element mentioning hip hop music or style but
doesn't discuss abything about beats or rythm is considered not
relevant. Also, elements discussing beats and rythm, but not
hip hop music in particular, are considered not
relevant.</narrative>
```

Figure 2.15 Query no 414

phrases are present (not relevant, not). But we admit that one can construct many examples where our strategy will not work, but such sentences are less frequent in such situations.

1.10.2 Algorithm

```
FOR all adjective and verb w
        calculate document frequency df and term frequency tf
        for w in positive and negative sentences
END FOR
sort list of words according to score(df,tf)
select n topmost words
FOR each sentence s in test data set
        no_of_occ = no of occurrence of negative words
        IF  no_of_occ %2 == 0
                put s in positive query
        ELSE
                put s in negative query
        END IF
END FOR
```

Figure 2.16 Algorithm for automatic negation detection

1.10.3 Result

We got 74% accuracy for this algorithm when applied to 65 INEX topics.

1.11 References

[1] Introduction to Information Retrieval by Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze

Future Work

- Information Retrieval
- XML Retrieval
- Our Work
- References

1.12 Element level retrieval

The goal of XML retrieval is to meet a user's information need at the sub-document level. But we could not extend our work to see the effect of the proposed work beyond document level. We have seen earlier in section 2.3.3 that one of the reasons for degradation of our result is the simultaneous occurrence of positive and negative terms in a document. Unfortunately we are not able to give any satisfactory solution to this problem. In case of sub-document level retrieval, chances of simultaneous occurrence of positive and negative keyword may reduce as the size of the retrieved unit reduces. In such a situation we expect our feedback model to give better result. This may be explored in future work.

1.13 Effective use of negation

Negative information in the query could not be used effectively. We only used simple subtraction in section 2.4.2 with same weighting given to both positive and negative terms. Effective tuning of their weighting based on previous relevance judged data could be another future extension to the current work.

1.14 Automatic detection of negation

Although we have proposed an algorithm to detect negative information from a XML query, this work can be extended to achieve more accurate negation detection. Some work has been done in this area [1, 2] but the approaches are designed to work for some specific domains like medical reports. It would be interesting to explore the extension of these approaches to the general domain.

1.15 References

[1] Implementation and Evaluation of Four Different Methods of Negation Detection by Sergey Goryachev, Margarita Sordo, Qing T. Zeng, Long Ngo.

[2] Negation Detection in Automated Medical Applications by Stefan Gindl.