Generating a Reference Set

A thesis
Submitted to the faculty of the graduate school
of the University of Minnesota
by

Kiran Kumar Bushireddy

In partial fulfillment of the requirements
for the degree of
Master of Science

Dr. Carolyn J. Crouch
September 2013

Acknowledgments

I would like to take this opportunity to sincerely thank Dr. Carolyn Crouch and Dr. Donald Crouch for providing me the platform to understand the science of Information Retrieval (IR), and inspiration in their domain expertise.

I would like to thank Dr. Joseph Gallian for teaching me a wonderful math course, which helped me to build my analytical skills, and for being a member of my thesis committee. He is one of the best mathematicians I have ever met in my life and his impact on me will be carried throughout the rest of my life.

I am grateful to Lori Lucia, Jim Luttinen, and Clare Ford for their continuous support in the completion of my thesis.

I thank Vamshi Krishna Thotempudi, a fellow graduate student who helped in proofreading my thesis write-up within a short time notice.

I am very thankful to my parents who gave me everything in life and for showing the path to lead an inspiring life.

Finally, I want to thank all my friends for their motivation and inspiration at tough times

Abstract

Information retrieval is the science of returning data from a corpus (a large collection of documents) matching the user's informational need. It identifies the data (originally in document form) by matching the terms in the query with terms contained in the documents of the collection. Representing documents and queries for effective retrieval is best accomplished by defining a model. Among the various models, the one most frequently used is Salton's Vector Space Model [9]. In this model, documents and queries are represented as vectors. The similarity between the query and a document is found by using a similarity measure (e.g., cosine).

Extensible Markup Language (XML) is a simple, flexible text format derived from Standard Generalized Markup Language (SGML) [3], designed to meet the challenges of large-scale electronic publishing, XML plays an important role in the exchange of a wide variety of data on the Web and elsewhere. INEX (*The Initiative for evaluation of XML retrieval*) [1] sponsors a competition that promotes the development of XML-based retrieval. It provides a Wikipedia collection in the form of XML files and each of these XML files are well defined and documented.

We are interested in building a reference run for a given set of queries without first performing a separate retrieval run to produce it. To that end, we perform some basic experiments which are based on the terminal nodes of the document set. The experiments depend upon the content of these nodes. Analysis of these early results, conclusions, and suggestions for future research are included.

Contents

List of Tables

List of Figures

1. Introduction

Information retrieval is the science of returning data from a corpus (a large collection of documents) matching the user's informational need. It identifies the data (originally in document form) by matching the terms in the query with terms contained in the documents of the collection. Representing documents and queries for effective retrieval is best accomplished by defining a model. Among the various models, the one most frequently used is Salton's Vector Space Model [9]. In this model, documents and queries are represented as vectors. The similarity between the query and a document is found by using a similarity measure (e.g., cosine). Smart [7] is an information retrieval system developed at Cornell University under the guidance of Gerard Salton, by Chris Buckley, and others. It uses the Vector Space Model.

Extensible Markup Language (XML) is a simple, flexible text format derived from Standard Generalized Markup Language (SGML) [3], designed to meet the challenges of large-scale electronic publishing. XML plays an important role in the exchange of a wide variety of data on the Web and elsewhere.

XML is a common method of representing documents, and the ability to retrieve various pieces of information (i.e., elements) from the XML structure has gained importance. INEX (*The Initiative for evaluation of XML retrieval*) [1] sponsors a competition that promotes the development of XML-based retrieval. The INEX competition provides a platform to work on various predefined tasks such as Social Book Search, Linked Data, Tweet Contextualization, Relevance Feedback and Snippet Retrieval. INEX provides a corpus and a set of user-generated queries along with metrics and evaluation tools that allow the evaluation and comparison of the systems designed by researchers to accomplish a specific task. The University of Minnesota Duluth (UMD) has an excellent track record of more than 10 years of participation with INEX and often ranks among the top 10 teams in terms of results. Most recently, the focus has been snippet retrieval.

A traditional retrieval system returns a document or reference to a document that matches the user query; such documents are typically unstructured. XML facilitates structured retrieval and retrieves data at a more granular level. We have built a structured retrieval system called Flex (Flexible retrieval system) to retrieve specific elements of a document. Flex is dynamic in nature; i.e., it retrieves elements at run time. Flex maintains an index of leaf nodes (e.g., paragraphs), and uses a representation of the document hierarchy (the doctree) to build the document from these nodes. Each element in the document tree is correlated with the query and a rank-ordered list of elements is generated for each tree.

Chapter 2 describes background for this work; Chapter 3 describes the basic task which we seek to accomplish and the experiments designed to facilitate that goal; Chapter 4 presents the results of these experiments and assesses their impact, and then concludes with suggestions for future work.

2. Background

This chapter presents the background material needed for this research. We begin with an overview of the Vector Space Model and the instantiation of the Vector Space Model via the Smart system and then describes dynamic element retrieval and Flex.

2.1 Vector Space Model

The Vector Space Model [9] is arguably the most effective model for retrieving information from a large collection of data. Both documents and queries are represented as term frequency vectors. For example, document D_j is represented as $D_j = (t_{1,j}, t_{2,j}, \ldots, t_{n,j})$, where t_i represents the frequency of term i in document D_j and n is the number of terms in the document. There are various methods to determine the weight of a term in the document. The Simplest methods use the count of the occurrences of a word in the document, known as term frequency. The similarity of a document with the query is established by producing the vector product of the two vectors using an appropriate similarity measure such as cosine or inner product, for example.

2.2 Smart

Smart [7] is an information retrieval system based on the Vector Space Model built by Chris Buckley and others at Cornell University under the guidance of Gerard Salton. Based on functionality Smart can be divided into three parts: namely, indexing, term weighting, and retrieval. We use Smart to index the terminal nodes of each document, producing a corresponding set of term-frequency vectors. User queries are also converted to vectors.

2.3 Inverted Index

Smart maintains an inverted index, a mapping of terms or concept ids to the documents in which that term is contained. It also contains the weight of that term in the document. A sample inverted file entry for concept id 34999947 is given in Figure 2.1.

Concept id	document id, weight		
34999947	9766085,1	11084418,1, ……,	29062748,1

Figure 2.1 Logical Representation of Inverted Index

2.4 Term Weighting

Smart assigns term frequency weights to terms. The three main components that affect the term weighting are term frequency (tf), inverse document frequency (idf), and document length normalization [8]. Smart allows the user to specify each of these three important components associated with term weighting, each represented as a single letter in three-letter word string. (The details of weighting schemes can be found in the Smart source library or at http://people.csail.mit.edu/jrennie/ecoc-svm/smart.html.)

Term frequency (tf) is the number of times a concept or word occurs in a particular document. Term frequency is used as the weight of the term in the first phase of the Smart indexing process. The higher the term frequency, the more important the term in that document. Inverse document frequency (idf) is the number of documents in which the term occurs. Longer documents have more terms and the likely repetition of terms, which improve their chances of being retrieved. To improve the retrieval chances in shorter documents, we use normalization (the third important component of term weighting). Document length normalization is a way of adjusting the term weights of a document in accordance with its length [10].

2.5 Flex

Traditional retrieval systems retrieve complete documents in response to a user's need whereas an element retrieval system retrieves specific element(s) from documents. Flex [5] is an element retrieval system developed at the University of Minnesota Duluth. This system takes as input the terminal node index produced by Smart and the doctree (or tree representation) of each document. Flex then generates each element of the document tree. Given the doctree and a set of terminal nodes for a document, Flex builds the whole

document, from terminal nodes to the top node of the tree. In this process, Flex correlates each node with the query and records the correlation score at each node. At the end of the process, Flex generates a list of elements in the document, rank-ordered by correlation with the query.

2.6 Terminal Node Set

A specific dump [4] of the Wikipedia collection is used in our experiments. The terminal nodes in this 2013 collection are paragraph (<p>), title (<title>) and header (<h>). The non-terminal nodes are abstract (<a>), section (<s>), page (<page>) and article (<xml>).

3. Basic Experiments

In this chapter, we describe the goal of our research and basic experiments.

3.1 Goal

Flex is an efficient element retrieval system. Given a reference run, a terminal node index for a set of documents and their doctrees, Flex generates the elements up to and including the documents themselves. However, consider the following case. Suppose a reference run is not available. Is it possible, given only a query, the doctree and the terminal node index, to determine which documents make up the reference run? This is the goal of our experiments - to ascertain if the reference run documents can be determined from the terminal node index alone, and if so, to what degree.

3.2 Relevance Assessments

INEX provides a 2013 Wikipedia collection in the form of XML files [4]. These files are parsed to produce the doctrees and the terminal nodes are indexed (as described in Chapter 2). INEX also provides queries and a reference run - a list of the top *n* documents retrieved by each query as per an official INEX retrieval run [4]. (INEX sets *n* to 20 for these experiments.) All the experiments in this chapter are performed using the terminal node index.

A group of graduate students from the Computer Science Department at the University of Minnesota Duluth did manual relevance assessments of the documents in the reference run to determine if these documents were in fact relevant. The results are given in Table 3.1.

Query	Total reference run documents	Relevant documents in reference run
1	20	5
2	20	7
3	20	11
4	20	2
5	20	3
6	20	0
7	20	8
8	20	8
9	20	6
10	20	5
11	20	15
12	20	4
13	20	8
14	20	13
15	20	16
16	20	12
17	20	8
18	20	14
19	20	4
20	20	4
21	20	8
22	20	14
23	20	13
24	20	10
25	20	13
26	20	3
27	20	2
28	20	18
29	20	1
30	20	6
31	20	20
32	20	1
33	20	1
34	20	6
35	20	5

Table 3.1 Manual Relevance Assessments (2013 Reference Run, Snippet Track)

3.3 Experiments

Section 3.3 discusses various experiments performed in this thesis.

Experiment 1: Using the Intersection and Union

A good document is one that contains all the key terms of query. This experiment is performed to find documents that contain all or any of the query terms using the Union-Intersection approach. A document is said to be in the intersection set if it contains every query term. A document is said to be in union set if it contains one or more of the query terms. Union serves as an upper bound on the number of documents that can be retrieved for a particular query. The algorithm to compute intersection and union sets is explained in Figure 3.1.

To evaluate our results, we determine whether a document is available in the reference run and also if the document has been assessed as relevant to the query. Results of this experiment along with the reference run and relevance assessments statistics are shown in Table 3.2.

For each query

For each term in the query

For each element in its inverted file entry

Identify the documents containing that element and form a list of these documents.

Form the union of the document sets.

Form the intersection of the document sets.

Figure 3.1 Algorithm to Produce the Document Union and Intersection Sets from Query Terms

	Intersection				Union			
Query	Total	In Ref run	Total relevant	Relevant Docs found	Total	In Ref run	Total relevant	Relevant Docs found
1	772	20	5	5	519953	20	5	5
2	9	7	7	7	7996	20	7	7
3	530	20	11	11	55464	20	11	11
4	177	9	2	2	177471	20	2	2
5	773	20	3	3	54264	20	3	3
6	17	11	0	0	2695	20	0	0
7	555	17	8	8	201376	20	8	8
8	44	20	8	8	9918	20	8	8
9	11	5	6	5	1769	19	6	6
10	1411	20	5	5	22438	20	5	5
11	1982	19	15	14	719747	20	15	15
12	102	4	4	3	167170	20	4	4
13	197	13	8	7	330753	20	8	8
14	110	20	13	13	670208	20	13	13
15	395	20	16	16	512208	20	16	16
16	218	0	12	0	143636	20	12	12
17	88	10	8	5	54835	20	8	8
18	3570	19	14	13	750668	20	14	14
19	722	5	4	0	349549	20	4	4
20	477	6	4	1	297380	20	4	4
21	29	3	8	1	773444	20	8	8
22	413	12	14	8	269806	20	14	14
23	210	9	13	4	257483	20	13	13
24	4874	20	10	10	136040	20	10	10
25	49	14	13	10	675995	20	13	13
26	330	18	3	2	322125	20	3	3
27	7	2	2	0	174879	20	2	2
28	25	12	18	12	130059	20	18	18
29	3	2	1	1	826361	20	1	1
30	1986	12	6	6	488391	20	6	6
31	1722	20	20	20	145768	20	20	20
32	440	20	1	1	20492	20	1	1
33	970	14	1	1	434560	20	1	1
34	230	20	6	6	79003	20	6	6
35	1377	4	5	1	357290	19	5	5

Table 3.2 Union-Intersection Sets with Respect to Reference Run and Relevance

In Table 3.2, column 1 represents the query id, column 2 the number of documents in the intersection, column 3 the number of reference run documents in the intersection, column 4 the number of reference run documents relevant to the query, column 5 the number of relevant documents in the intersection, column 6 the total number of documents in the union, column 7 the number of reference run documents in the union, column 8 the number of relevant documents in the reference run, and column 9 is the number of relevant documents in the union.

For an example, query 14 has 110 documents in the intersection and 20 reference run documents are present in it. Thirteen of the 20 reference documents are judged relevant and all are present in the intersection. Query 14 has 670,208 documents in the union; all 20 reference run documents are present in it. All 13 of the 20 reference documents judged relevant are present in the union.

Experiment 2: Using Proper Nouns and Phrases as Filters

Table 3.2 shows numbers of the documents in the intersection and in the union. Observing the queries shows the phrases and the proper nouns present in them. The goal of this experiment is to filter out the documents that do not contain these entities. The objective is to reduce the size of the union and intersection without losing documents related to query. Thus, we filter the text of each document in the union and intersection. Results are shown in Table 3.3.

In Table 3.3, column 1 is the query id, column 2 is the total number of documents in the intersection after filtering (if column 6 is empty, no such filters are present in the query), column 3 is the number of reference run documents present in the intersection after filtering, column 4 is the total number of documents in the union after filtering, column 5 is the number of reference run documents present in the union after filtering, and column 6 contains the phrase or proper noun used to filter the query.

For example, after filtering the document sets with the phrase “John Lennon”, query 1 has 636 documents in the intersection and 17 of the 20 reference run documents are present in it. After filtering, the union contains 2411 documents, and 17 of the total 20 reference documents are present in it.

	Intersection		Union		
Query	Total reduction in intersection	Reduction in ref run	Total reduction in intersection	Reduction in ref run	Phrase
1	636/772	17/20	2411/519953	17/20	“John Lennon”
2	9/9	7/7	12/7996	7/7	“Mega Bloks”
3	530	20	55464	20	
4	177	9	177471	20	
5	773	20	54264	20	
6	17	11	2695	20	
7	314/555	16/17	2959/201376	19/20	“Crystal Palace”
8	44	20	9918	20	
9	11	5	1769	19	
10	1411	20	22438	20	
11	126/1982	18/19	424/719747	19/20	“Regular expressions”
12	102	4	167170	20	
13	197	13	330753	20	
14	110	20	670208	20	
15	395	20	512208	20	
16	218	0	143636	20	
17	88	10	54835	20	
18	215/3570	13/19	1404/750668	14/20	“Police cars”
19	78/722	2/5	626/349549	7/20	“Chemical elements”
20	477	6	297380	20	
21	28/29	3/3	1160/773444	20/20	“Marilyn Monroe”
22	43/413	12/12	607/269806	20/20	“Gallo roman”
23	10/210	0/9	70/257483	0/20	“Flute solos”
24	175/4874	17/20	175/136040	17/20	“Bond girls”
25	37/49	14/14	233/675995	19/20	“Kargil war”
26	330	18	322125	20	
27	7	2	174879	20	
28	25	12	130059	20	
29	3/3	2/2	26/826361	8/20	“Tose Proeski”
30	1986	12	488391	20	
31	3/1722	0/20	6/145768	0/20	“Euro vision”
32	2/440	0/20	2/20492	0/20	“Alien android”
33	3/970	1/14	312/434560	1/20	“Hand injury”
34	58/230	16/20	182/79003	16/20	“Demonic possession”
35	9/1377	0/4	187/357290	0/19	“Australian research”

Table 3.3 Union-Intersection Filtering (Phrases and Proper Nouns) with Respect to Reference Run

We expect reduction in the size of both the intersection and union after filtering. Consider query 7 with 17 of 20 reference run documents in the intersection of 555 and a union of 201,376. After filtering, the intersection reduces to 314 with 16 of 20 reference documents in it. The union is significantly decreased from 201,376 to 2959 and 19 of the 20 reference documents are present in it. Reducing the intersection from 555 to 314 while losing 1 reference document has no impact on the number of relevant documents [i.e., we have all 8 relevance documents in the reference run even after filtering using proper nouns and phrases, as seen in Table 3.4.]

In Table 3.4, column 1 is the query id, column 2 is the reduction of intersection from Experiment 1 to this experiment, column 3 is the number of relevant documents found in this experiment compared to total number of relevant documents in the reference run and column 4 contains the phrase or proper noun used to filter the query. Consider query 1. There is a reduction in intersection from 772 to 636 documents and all 5 of the relevant documents in reference run are present in it.

Experiment 3: Using Flex to Rank Documents in Intersection

After performing Experiment 2, we observe that both the size of union and of the intersection are reduced but an appropriate ranking mechanism to retrieve the top-ranked documents from these sets is not available.

As an input to Flex, both intersection and union can be seeded. But in this experiment in an effort to retrieve top-ranked documents, we seed only the intersection to Flex, since union contains too many documents to be of use. After seeding, Flex produces correlation scores for elements at all levels in the document, including the root level, and for all seeded documents. Finally, based on these scores, documents are sorted in descending order to obtain the top *n*-ranked documents. For more information about Flex and its operation, see [2].

This experiment is performed with two different input sets: 1) Seeding the intersection directly, and 2) seeding the intersection after applying phrase filtering. Results of the first approach are shown in Table 3.5 and of the second in Table 3.6.

Query	Reduction in intersection	Found Rel/ Total Rel	Phrase
1	636/772	5/5	“John Lennon”
2	9/9	7/7	“Mega Bloks”
3	530	11/11	
4	177	2/2	
5	773	3/3	
6	17	0/0	
7	314/555	8/8	“Crystal Palace”
8	44	8/8	
9	11	5/6	
10	1411	5/5	
11	126/1982	14/15	“Regular expressions”
12	102	3/4	
13	197	7/8	
14	110	13/13	
15	395	16/16	
16	218	0/12	
17	88	5/8	
18	215/3570	9/14	“Police cars”
19	78/722	0/4	“Chemical elements”
20	477	1/4	
21	28/29	1/8	“Marilyn Monroe”
22	43/413	8/14	“Gallo roman”
23	10/210	0/13	“Flute solos”
24	175/4874	10/10	“Bond girls”
25	37/49	10/13	“Kargil war”
26	330	2/3	
27	7	0/2	
28	25	12/18	
29	3/3	1/1	“Tose Proeski”
30	1986	6/6	
31	3/1722	20/20	“Euro vision”
32	2/440	0/1	“Alien android”
33	3/970	1/1	“Hand injury”
34	58/230	6/6	“Demonic possession”
35	9/1377	0/5	“Australian research”

Table 3.4 Intersection Filtering (Phrases and Proper Nouns) with Respect to Relevance

Query	Ref docs in intersection	Relevant Docs in intersection	Total Docs in intersection	Number of top-ranked docs required to retrieve all Ref (and Rel docs)
1	20	5/5	772	78
2	7	7/7	9	7
3	20	11/11	530	210
4	9	2/2	177	38
5	20	3/3	773	129
6	11	0/0	17	14
7	17	8/8	555	87
8	20	8/8	44	35
9	5	5/6	11	6
10	20	5/5	1411	183
11	19	14/15	1982	464
12	4	3/4	102	7
13	13	7/8	197	130
14	20	13/13	110	89
15	20	16/16	395	76
16	0	0/12	218	1
17	10	5/8	88	61
18	19	13/14	3570	778
19	5	0/4	722	24
20	6	1/4	477	75
21	3	1/8	29	7
22	12	8/14	413	341
23	9	4/13	210	27
24	20	10/10	4874	91
25	14	10/13	49	31
26	18	2/3	330	74
27	2	0/2	7	2
28	12	12/18	25	21
29	2	1/1	3	2
30	12	6/6	1986	326
31	20	20/20	1722	518
32	20	1/1	440	118
33	14	1/1	970	287
34	20	6/6	230	189
35	4	1/5	1377	489

Table 3.5 Seeding Intersection to Flex with Respect to Reference and Relevant Documents

Query	Ref docs in intersection after phrase filtering	Rel docs in intersection after phrase filtering	Total docs in intersection after phrase filtering	Number of top-ranked docs required to retrieve all Ref and Rel docs (after filtering)
1	17	5/5	636	69
2	7	7/7	9	7
3	20	11/11	530	210
4	9	2/2	177	38
5	20	3/3	773	129
6	11	0/0	17	14
7	16	8/8	314	62
8	20	8/8	44	35
9	5	5/6	11	6
10	20	5/5	1411	183
11	18	14/15	126	59
12	4	3/4	102	7
13	13	7/8	197	130
14	20	13/13	110	89
15	20	16/16	395	76
16	0	0/12	218	0
17	10	5/8	88	61
18	13	9/14	215	72
19	2	0/4	78	4
20	6	1/4	477	75
21	3	1/8	28	7
22	12	8/14	43	41
23	0	0/13	10	0
24	17	10/10	175	53
25	14	10/13	37	25
26	18	2/3	330	74
27	2	0/2	7	2
28	12	12/18	25	21
29	2	1/1	3	2
30	12	6/6	1986	326
31	0	20/20	3	0
32	0	0/1	2	0
33	1	1/1	3	1
34	16	6/6	58	56
35	0	0/5	9	0

Table 3.6 Seeding Filtered Intersection to Flex with Respect to Reference and Relevant Documents

In Table 3.5, column 1 represents query id, column 2 the number of reference run documents in intersection set, column 3 the number of relevant documents in the intersection, column 4 is the total number of documents in intersection set, column 5 is the number of top *n* ranked documents retrieved for each query to get all the reference run documents. Column labels of Table 3.6 are identical to those in Table 3.5 except the input set is unfiltered.

With approach 1 (seeding the intersection directly) as input to Flex, retrieving the top 130 ranked documents results in all of the reference run documents in intersection for 25 queries out of 35, and with approach 2 (seeding filtered intersection) as input, it retrieves all reference run documents in the filtered intersection for 32 queries out of 35.

Consider query 11. The intersection has 1982 documents; 19 of 20 reference documents and 14 of 15 relevant documents are in it. By seeding this intersection and retrieving top 464 documents from Flex, we get all 19 reference documents and 14 relevant documents in intersection.

Experiment 4: Using the Description as the Query

Every query is divided into three portions: title, description, and narrative. As part of the INEX task, only the title portion of each query is indexed. In this experiment, the description portion of each query is indexed instead of the title to see if it produces improved results from the union and intersection. Results are included in Table 3.7.

Column labels of Table 3.7 are identical to those in Table 3.2. Consider query 11 in Table 3.7. The intersection contains 1635 documents including 17 of 20 reference run documents and 12 of the 15 relevant reference documents. The union contains 946,732 documents, including the 20 reference documents and all 15 of the relevant documents.

Comparing results of Experiments 1 and 4 (looking at each query with statistics provided) shows that Experiment 1 produces better results in all cases. Observing query 1 statistics in both the experiments gives us a better understanding of performance. Experiment 1 uses the query title “Death of John Lennon” and produces an intersection of 772 documents with all 20 reference run documents in it.

	Intersection				Union			
Query	Total	In Ref run	Total Rel	Rel docs in Experiment	Total	In Ref run	Total Rel	Rel Docs in Experiment
1	212/772	7/20	5	3/5	687634/519953	20/20	5/5	5/5
2	0/9	0/7	7	0/7	564198/7996	20/20	7/7	7/7
3	530/530	20/20	11	11/11	55464/55464	20/20	11/11	11/11
4	55/177	2/9	2	2/2	398359/177471	20/20	2/2	2/2
5	239/773	7/20	3	3/3	559011/54264	20/20	3/3	3/3
6	11/17	5/11	0	0/0	348779/2695	20/20	0/0	0/0
7	182/555	4/17	8	4/8	402663/201376	20/20	8/8	8/8
8	21/44	11/20	8	7/8	602592/9918	20/20	8/8	8/8
9	1/11	0/5	6	0/5	715582/1769	19/19	6/6	6/6
10	263/1411	6/20	5	2/5	123216/22438	20/20	5/5	5/5
11	1635/1982	17/19	15	12/14	946732/719747	20/20	15/15	15/15
12	68/102	2/4	4	2/3	446123/167170	20/20	4/4	4/4
13	69/197	5/13	8	1/7	328577/330753	20/20	8/8	8/8
14	63/110	14/20	13	9/13	858351/670208	20/20	13/13	13/13
15	173/395	15/20	16	13/16	712181/512208	20/20	16/16	16/16
16	286/218	0/0	12	0/0	573554/143636	20/20	12/12	12/12
17	4/88	1/10	8	1/5	760589/54835	20/20	8/8	8/8
18	6119/3570	19/19	14	13/13	321481/750668	20/20	14/14	14/14
19	467/722	2/5	4	0/0	677716/349549	20/20	4/4	4/4
20	320/477	5/6	4	1/1	732063/297380	20/20	4/4	4/4
21	19/29	1/3	8	0/1	836163/773444	20/20	8/8	8/8
22	413/413	12/12	14	8/8	269806/269806	20/20	14/14	14/14
23	210/210	9/9	13	4/4	257483/257483	20/20	13/13	13/13
24	98/4874	2/20	10	2/10	1082723/136040	20/20	10/10	10/10
25	1/49	0/14	13	0/10	1090697/675995	20/20	13/13	13/13
26	74/330	1/18	3	0/2	470430/322125	20/20	3/3	3/3
27	1/7	0/2	2	0/0	596307/174879	20/20	2/2	2/2
28	77/25	5/12	18	5/12	356904/130059	20/20	18/18	18/18
29	0/3	0/2	1	0/1	1021726/826361	20/20	1/1	1/1
30	276/1986	2/12	6	2/6	573150/488391	20/20	6/6	6/6
31	32/1722	0/20	20	0/20	1186854/145768	20/20	20/20	20/20
32	57/440	1/20	1	0/1	473051/20492	20/20	1/1	1/1
33	197/970	0/14	1	0/1	607074/434560	20/20	1/1	1/1
34	22/230	1/20	6	1/6	176231/79003	20/20	6/6	6/6
35	171/1377	0/4	5/5	0/1	475973/357290	19/19	5/5	5/5

Table 3.7 Union-Intersection Approach Using the Description as the Query

Experiment 4 uses the description "Information about John Lennon's death" and produces an intersection of 212 documents with only 7 of the 20 reference documents in it. Though the number of documents in the union (1 and 4) varies, both contain all 20 of the reference run documents. It seems that in Experiment 4 query 1 the additional terms in the description (intended to clarify it) have unintended results. E.g., the word "information" is not specific to John Lennon and greatly increases the size of the corresponding document sets.

Experiment 5: Using Automatically Generated Phrases from the Description

Results of Experiment 4 (as shown in Table 3.7) indicate increased numbers of documents in the union. Similar to Experiment 2, here we apply filtering to reduce the number of documents in the union and to generate documents containing important phrases related to query. This experiment differs from Experiment 2 in that it uses machine-generated phrases rather than manually selected phrases.

MontyLingua [6] is a free, commonsense enriched, end-to-end natural language understanding tool that extracts noun phrases, verb phrases, people's names, places, events and other semantic information. Phrases generated from this tool are used in this experiment. The algorithm is the same as that in Experiment 2. Results of this experiment along with information about the number of documents in the reference run are shown in Table 3.8.

The description of each column in Table 3.8 is given in Experiment 2. For example, consider query 11. It has 63 documents in the intersection and each of these documents contains either the phrase "Text editor" or "Regular expression support" along with all query terms. 15 of the 20 reference run documents are present in the intersection. The Union contains 764 documents and 17 documents of 20 reference documents are present in it. If there are multiple phrases for a query, documents in the union are filtered using one of the phrases.

	Intersection set		Union set		
Query	Total	In Ref Run	Total	In Ref Run	Phrase used for filtering
1	130	2	182804	2	(Information)|(John Lennon's death)
2	0	0	182804	2	(Information)|(Lego vs mega bloks Law suits)
3	2	0	2	0	Banana diseases
4	55	2	54623	20	(Safety)|(SUVs)|(Regular cars)
5	152	7	22876	20	(Sharp)|(Apple products)
6	0	0	82517	0	(MRI and cyclotron)|(relationship)
7	127	3	182804	3	(Information)|(Crystal palace fire)
8	21	11	132587	20	(What)|(penguins)|(Dunedin)
9	1	0	170828	19	(What)|(Gelatin)|(BSE)|(it)
10	257	6	84787	20	(What)|(difference)|(thesis)|(dissertation)
11	63	15	764	17	(Text editors)|(Regular expressions support)
12	68	2	24142	20	(Controversial episodes)|(TV series)| (Seinfeld)
13	14	0	21939	0	(Countries)|(rainforests and deserts)
14	20	13	1211	15	(Nintendo video games)|(which luigi) |(playable character)
15	173	15	294193	16	(Public transport systems)|(smart card) |(systems?)
16	47	0	1586	0	(Animated Disney films?) |(Princesses)
17	4	1	384468	20	(Articles)|(suggested ratio) |(carbohydrates) |(Fat and protein)|(human diet)
18	6038	19	105529	20	(Cars)|(Police vehicles)
19	460	1	426442	1	(People)| (multiple chemical elements)
20	314	5	427099	20	(People)|(banknote)
21	4	0	7089	3	(Marilyn Monroe impersonator or actor)| (her role)
22	413	12	120309	20	(Gallo)|(architecture)|(Paris)
23	50	2	2211	4	(Famous flute solos)|(Symphonies)
24	98	2	400307	15	(James bond girls)|(James bond movies)|(actresses)|(name)|(role)
25	1	0	561	19	(Indo Pakistan relations)|(Kargil war)|(particular reference)
26	74	1	230039	20	(Beer)|(health)|(effects)
27	1	0	182804	6	(Health implication)| (information)| (traditional coca leaf consumption)
28	44	5	12819	17	(Volcanic ash)|(planes)
29	0	0	0	0	(Tose Proeski's humanitarian work)

	Intersection set		Union set		
Query	Total	In Ref Run	Total	In Ref Run	Phrase used for filtering
30	276	2	471308	20	(Adult human height)| (factors)|(That)
31	32	0	141652	20	(Eurovision song contest)| (voting work) |(countries)|(blocks)
32	46	0	182814	2	(Information)|(movie alien)
33	22	0	2907	0	(Medical specialist)|(my hand injury)|(what kind)
34	21	1	8428	19	(My neighbor)|(demon)|(exorcism)
35	171	0	213504	13	(Money)|(excellent researchers)|(Australia)

Table 3.8 Union-Intersection Approach Using Description as the Query, Automatic Phrase Filtering with Respect to Reference

Observing Tables 3.7 and 3.8, we see union has been largely reduced. For query 4, the union is reduced from 398,359 to 54,623 and retains all 20 of the reference documents (an upper bound is too large to be useful). For most of the queries, the union is reduced at the expense of losing reference run documents. In query 1 from Experiment 4 we have a union of 687,634 documents with all 20 reference run documents in it. After filtering using this phrase technique, the union is reduced to 182,804 documents, with only 2 documents out of the total 20 reference run documents in it. This same scenario is observed for approximately 1/3 of the queries (i.e. 13 of 35).

The description of each column in Table 3.9 is given in Experiment 2. For example, consider query 11. It has 63 documents in the intersection; 11 of 15 relevant documents are in the intersection.

Analyzing the results of Experiments 2 and 5 leads to the conclusion that union and intersection formed using query terms from the title perform better than those from the description. Most of the machine-generated phrases are not present in the documents of union and intersection sets.

Experiment 6: Using the Narrative

In Experiment 4, we indexed the description of the query rather than its title, whereas in this experiment we index the narrative portion of query as query terms.

Query	Total	Found Rel / Total Rel	Phrase used for filtering
1	130	2/5	(Information)|(John Lennon's death)
2	0	0/7	(Information)|(Lego vs mega bloks Law suits)
3	2	0/11	Banana diseases
4	55	2/2	(Safety)|(SUVs)|(Regular cars)
5	152	3/3	(Sharp)|(Apple products)
6	0	0/0	(MRI and cyclotron)|(relationship)
7	127	3/8	(Information)|(Crystal palace fire)
8	21	7/8	(What)|(penguins)|(Dunedin)
9	1	0/6	(What)|(Gelatin)|(BSE)|(it)
10	257	2/5	(What)|(difference)|(thesis)|(dissertation)
11	63	11/15	(Text editors)|(Regular expressions support)
12	68	2/4	(Controversial episodes)|(TV series)| (Seinfeld)
13	14	0/8	(Countries)|(rainforests and deserts)
14	20	9/13	(Nintendo video games)|(which luigi)| (playable character)
15	173	13/16	(Public transport systems)|(smart card) |(systems?)
16	47	0/4	(Animated Disney films?) |(Princesses)
17	4	1/8	(Articles)|(suggested ratio) |(carbohydrates) |(Fat and protein)|(human diet)
18	6038	13/14	(Cars)|(Police vehicles)
19	460	0/4	(People)| (multiple chemical elements)
20	314	1/4	(People)|(banknote)
21	4	0/8	(Marilyn Monroe impersonator or actor)| (her role)
22	413	8/14	(Gallo)|(architecture)|(Paris)
23	50	0/13	(Famous flute solos)|(Symphonies)
24	98	2/10	(James bond girls)|(James bond movies)| (actresses)| (name)| (role)
25	1	0/13	(Indo Pakistan relations)|(Kargil war)|(particular reference)
26	74	0/3	(Beer)|(health)|(effects)
27	1	0/2	(Health implication)| (information)| (traditional coca leaf consumption)
28	44	5/18	(Volcanic ash)|(planes)
29	0	0/1	(Tose Proeski's humanitarian work)
30	276	2/6	(Adult human height)| (factors)|(That)
31	32	0/20	(Eurovision song contest)| (voting work) |(countries)|(blocks)
32	46	0/1	(Information)|(movie alien)
33	22	0/1	(Medical specialist)|(my hand injury)|(what kind)
34	21	1/6	(My neighbor)|(demon)|(exorcism)
35	171	0/5	(Money)|(excellent researchers)|(Australia)

Table 3.9 Intersection Using Description as the Query, Automatic Phrase Filtering with Respect to Relevance

The goal of this experiment is to determine whether indexing the narrative associated with the query produces better results. The results of this experiment along with reference run and relevant documents are shown in Table 3.10.

Consider query 11. It has 235 documents in the intersection and none of these documents are present in reference run. There are 15 relevant documents in the reference run but none of them are present in the intersection. The union has 1,230,299 documents including all 20 reference run documents and the 15 relevant documents.

From Table 3.10, we see that most of the intersections are very small and don't contain a majority or most of the reference run documents. Consider the impact of the lengthening of the query. For example, consider query 1 with narrative tag "I want to know how where and when (including time of day) when John Lennon died. Now I know he was shot, but what was the name of the guy who shot him? " It generated an intersection of 44 documents with only 1 of the 20 reference run documents present in it. As the number of query terms increase, the size of the intersection set gradually decreases and most of these documents are not present in the reference run.

Experiment 7: Using Automatically Generated Phrases from the Narrative

This experiment uses the automatically generated phrases by MontyLingua tool from the narrative provided with the queries. Results of this experiment along with information about the number of documents in the reference run are shown in Table 3.11.

Consider query 10. It has an intersection of 39 documents and only 1 of the 20 reference documents is present in it. The union has 126,156 documents and all 20 reference run documents are in it. Every document in this union contains one of the phrases shown in Table 3.11 for query 10.

The intersection from Table 3.10 is further reduced (after applying phrase search for query 11) from 235 to 18 as seen in Table 3.11. For 17 queries in Table 3.10, the intersection is empty. Similarly the union is reduced for few queries at the cost of losing most of the reference run documents. We observe that Experiment 2 produces better

	Intersection				Union			
Query	Total	In Ref run	Total relevant	Number of Relevant Docs	Total	In Ref run	Total relevant	Relevant found in experiment
1	44	1	5	1	1762861	20	5	5
2	0	0	7	0	579112	20	7	7
3	1	0	11	0	1001250	20	11	11
4	2	0	2	0	1015729	20	2	2
5	27	0	3	0	668303	20	3	3
6	0	0	0	0	745583	20	0	0
7	0	0	8	0	981438	20	8	8
8	0	0	8	0	1450546	20	8	8
9	0	0	6	0	1447382	20	6	6
10	39	1	5	0	229617	20	5	5
11	235	0	15	0	1230299	20	15	15
12	24	2	4	2	739346	20	4	4
13	1	0	8	0	1067054	20	8	8
14	0	0	13	0	1561799	20	13	13
15	18	0	16	0	1794808	20	16	16
16	2	0	12	0	922956	20	12	12
17	2	0	8	0	1086371	20	8	8
18	25	0	14	0	1211145	20	14	14
19	133	0	4	0	1529936	20	4	4
20	38	0	4	0	935930	20	4	4
21	0	0	8	0	1762798	20	8	8
22	5	0	14	0	1340312	20	14	14
23	1	0	13	0	1521329	20	13	13
24	24	0	10	0	1251181	20	10	10
25	0	0	13	0	1971173	20	13	13
26	58	2	3	0	947717	20	3	3
27	0	0	2	0	483531	20	2	2
28	0	0	18	0	967111	20	18	18
29	0	0	1	0	2425044	20	1	1
30	0	0	6	0	1832153	20	6	6
31	1	0	20	0	1585446	20	20	20
32	0	0	1	0	931660	20	1	1
33	25	0	1	0	796562	20	1	1
34	0	0	6	0	630224	20	6	6
35	8	0	5	0	875418	19	5	5

Table 3.10 Union-Intersection Approach Using the Narrative as the Query with Respect to Reference Run and Relevance

results in terms of the number of documents in the intersection and union and also produces more reference documents in these sets.

The description of each column in Table 3.12 is identical to Experiment 2. For example, consider query 11. It has 18 documents in the intersection and none of 15 relevant documents are in the intersection.

	Intersection		Union		
Query	Total	In reference run	Total	In reference run	Phrase used for filtering
1	44	1	942339	20	(Time)|(day)|(John Lennon)
2	0	0	17777	0	(Outcome)|(Lego vs mega bloks law suits?)
3	1	0	445606	11	(There)|(several diseases)|(banana populations)|(Past leaving)
4	0	0	65	2	(Usual SUVs)|(Regular cars)
5	6	0	4475	0	(Sharp manufacture)|(apple products)|(Which ones?)
6	0	0	2186	11	(MRI)|(my hip)
7	0	0	33919	19	(Invention)|(Crystal palace)|(cool photos)|(Photography)
8	0	0	627692	20	(ADCS conference)|(year)|(Dunedin)
9	0	0	253114	15	(What)|(Gelatin)|(BSE)
10	39	1	126156	20	(What)|(difference)|(thesis)|(dissertation)
11	18	0	1146	19	(Text editors)|(Regular expressions)
12	24	2	103041	20	(Episodes)|(TV series)|(Seinfeld)
13	1	0	160121	14	(Countries)|(wide range)|(climates)|(Rainforests and deserts)
14	0	0	646749	20	(Video game character)|(world)|(Luigi)|(Mario)|(brother)
15	18	0	561515	20	(Public transport systems)|(world)|(smart card)|(recent years)
16	0	0	1586	0	(Animated Disney films)|(Princesses)
17	2	0	36160	20	(Three major nutrients)|(humans)|(health problems)|(carbohydrates)|(Fats and proteins)|(Their diets)
18	23	0	292281	18	(Police cars)|(Police force)|(Regular passenger vehicles)|(Use)

	Intersection		Union		
Query	Total	In reference run	Total	In reference run	Phrase used for filtering
19	80	0	32736	16	(118 known chemical elements)|(past 300 years)|(Element)
20	38	0	433396	20	(People)|(banknote)|(currency)
21	0	0	374696	20	(Marilyn Monroe)|(several actors)|(her legacy)|(different perceptions)|(role)|(book)
22	5	0	163907	18	(Gallo roman architecture ruins)|(buildings)| (Paris)
23	1	0	30989	15	(Primary flute)|(orchestra)
24	24	0	316255	18	(Bond girls)|(Corresponding movie)|(actresses)|(article)|(role)|(biography)
25	0	0	35982	1	(Documents)|(post independence relationship)|(two neighboring countries)
26	58	2	294490	20	(Information)|(benefits)|(beer)|(health)
27	0	0	4474	19	(Coca)|(drinking coca tea)|(drinking coffee or tea)
28	0	0	182806	8	(Information)|(volcanic ash clouds)|(flight disruptions or incidents)
29	0	0	40	8	(Tose Proeski)|(popular Macedonian Singer)| (entire Balkan)
30	0	0	288140	5	(Members)|(my family and relatives)|(Their age)|(gender)|(ethnicity)
31	0	0	4809	20	(Eurovision song contest)|(voting process)
32	0	0	349525	4	(Information)|(movie alien)|(Ridley Scott)|(role)
33	8	0	965	0	(My keyboard)|(my hand)
34	0	0	7772	15	(Scary neighbor)|(demon)
35	8	0	289377	18	(Money)|(researchers)|(Australia)|(lot)

Table 3.11 Union-Intersection Approach Using the Narrative as the Query, Automatic Phrase Filtering with Respect to Reference

Query	Total	Found Rel/ Total Rel	Phrase used for filtering
1	44	1/5	(Time)|(day)|(John Lennon)
2	0	0/7	(Outcome)|(Lego vs mega bloks law suits?)
3	1	0/11	(There)|(several diseases)|(banana populations)|(Past leaving)
4	0	0/2	(Usual SUVs)|(Regular cars)
5	6	0/3	(Sharp manufacture)|(apple products)|(Which ones?)
6	0	0/0	(MRI)|(my hip)
7	0	0/8	(Invention)|(Crystal palace)|(cool photos)|(Photography)
8	0	0/8	(ADCS conference)|(year)|(Dunedin)
9	0	0/6	(What)|(Gelatin)|(BSE)
10	39	0/5	(What)|(difference)|(thesis)|(dissertation)
11	18	0/15	(Text editors)|(Regular expressions)
12	24	2/4	(Episodes)|(TV series)|(Seinfeld)
13	1	0/8	(Countries)|(wide range)|(climates)|(Rainforests and deserts)
14	0	0/13	(Video game character)|(world)|(Luigi)|(Mario)|(brother)
15	18	0/16	(Public transport systems)|(world)|(smart card)|(recent years)
16	0	0/12	(Animated Disney films)|(Princesses)
17	2	0/8	(Three major nutrients)|(humans)|(health problems)|(carbohydrates)|(Fats and proteins)|(Their diets)
18	23	0/14	(Police cars)|(Police force)|(Regular passenger vehicles)|(Use)
19	80	0/4	(118 known chemical elements)|(past 300 years)|(Element)
20	38	0/4	(People)|(banknote)|(currency)
21	0	0/8	(Marilyn Monroe)|(several actors)|(her legacy)|(different perceptions)|(role)| (book)
22	5	0/14	(Gallo roman architecture ruins)| (buildings)| (Paris)
23	1	0/13	(Primary flute)|(orchestra)
24	24	0/10	(Bond girls)|(Corresponding movie)| (actresses)| (article)|(role)|
25	0	0/13	(Documents)|(post independence relationship)| (two neighboring countries)
26	58	0/3	(Information)|(benefits)|(beer)|(health)
27	0	0/2	(Coca)|(drinking coca tea)|(drinking coffee or tea)
28	0	0/18	(Information)|(volcanic ash clouds)| (flight disruptions or incidents)
29	0	0/1	(Tose Proeski)|(popular Macedonian Singer)| (entire Balkan)
30	0	0/6	(Members)|(my family and relatives)| (Their age)| (gender)|
31	0	0/20	(Eurovision song contest)|(voting process)
32	0	0/1	(Information)|(movie alien)|(Ridley Scott)|(role)
33	8	0/1	(My keyboard)|(my hand)
34	0	0/6	(Scary neighbor)|(demon)
35	8	0/5	(Money)|(researchers)|(Australia)|(lot)

Table 3.12 Intersection Using Narrative as the Query, Automatic Phrase Filtering with respect to Relevance

Experiment 7 generates a very small intersection. Filtering the input using automatically generated phrases further reduces it; for 17 of 35 queries, the intersection is empty. This experiment is analyzed with respect to reference run and relevance and finds only 6 of 700 reference documents and only 3 of 274 relevant documents.

Using the narrative fails to identify documents related to the query. MontyLingua output must be analyzed to identify useful and meaningful phrases and only these should be applied. An unimportant phrase simply reduces the intersection, and other documents related to query are lost in filtering.

4. Experimental Results, Conclusions, and Suggestions for Future Research

Section 4.1 gives a brief overview of our experimental results, and Section 4.2 gives conclusions and suggestions for future research.

4.1 Experimental Results

Experiment 1 calculates the intersection and union of documents which contain query terms indexed from the title portion of queries. Documents in the intersection and union are examined to see if they are present in the INEX reference run and are also checked for relevance. Input to this experiment is terminal node index. For 25 queries, the intersection contains more than 100 documents. This experiment produced 447 of the 700 reference documents and 209 of 274 relevant documents.

Experiment 2 is built on the result of Experiment 1. Proper nouns and phrases present in the queries are applied as filters on union and intersection in an effort to reduce the size of sets without losing documents related to query. The result is unions and intersections with documents containing these phrases in them. Each document is then compared with reference run documents and relevance assessments. A total of 360 documents of 700 reference run documents were produced with a loss of 87 documents due to filtering, and 199 of 274 relevant documents were produced with a loss of 10 relevant documents when compared to the corresponding results in Experiment 1.

Experiment 3 ranks the documents in the intersection based on two approaches using Flex. In the first approach, results of Experiment 1 (i.e., plain intersection) are seeded and the top-ranked *n* documents are retrieved so as to get all the reference run documents from the input. In the second approach, results of Experiment 2 (i.e., intersection after applying phrase filtering) are taken as the input, and the top-*n* ranked documents are similarly retrieved. We note that retrieving the top 130 documents from Flex results in capturing all reference run documents for 25 of 35 queries in the first approach and 32 of 35 queries in the second approach.

In Experiment 4, the description is indexed as the query instead of title, and both union and intersection are formed as before. This experiment is similar to Experiment 1 and produces 176 of 700 reference run documents and 106 of 274 relevant documents. Additional terms in the description field yield unintended results; thus, this experiment produces worse results than Experiment 1. Moreover, extra terms in the query lead to smaller intersections, bigger unions, and a reduction in the number of reference run documents.

Experiment 5 takes the result of Experiment 4 as input and filters the union and intersection using phrases generated by MontyLingua [6]. Some of the phrases generated are not useful in the context of the query and as a result, only 127 of 700 reference run documents and 87 of 274 relevant documents are generated. Comparing these results to those of Experiment 2 shows that Experiment 2 performed better.

Experiment 6 is similar to Experiments 2 and 4 in finding union and intersection except that it uses the narrative portion of the query as the query itself. In general, the narrative contains more terms than title and description. More query terms reduce the intersection and the chance of finding reference run and relevant documents in it. Only 6 of 700 reference documents and 3 of 274 relevant documents were produced in this intersection. This experiment produces poor results when compared to Experiments 1 and 4.

Experiment 7 takes the result of Experiment 6 as its input and applies automatic phrases generated by MontyLingua [6]. As the input to this experiment has few reference run or relevant documents, positive results cannot be expected.

Experiments conducted in Chapter 3 are evaluated using 2 measures: 1) the number of relevant documents in the intersection and 2) the number of reference documents in intersection. A summary of results of these experiments, performed using title, description and narrative, is given in Table 4.1. Results show that experiments performed using title produced better results than description and narrative.

Experiment	Total Reference run docs for query set	Total Relevant docs for query set	Query Terms Indexed from
1	447/700	209/274	Title
2	360/700	199/274	Title
4	176/700	106/274	Description
5	127/700	87/274	Description
6	6/700	3/274	Narrative
7	6/700	3/274	Narrative

Table 4.1 Summary of Experiments with Respect to Reference Run and Relevance

Analyzing the results of Experiment 1, we can identify 3 sets of queries: 1) outstanding 2) average and 3) poor. The outstanding group has more than 17 of 20 reference documents for each query (i.e., 85% of reference documents are present in intersection) and contains 15 of the 35 queries. The average group has 10 to 17 reference documents for each query; there are 8 of them. The poor group has fewer than 50% of the reference documents for each query, and there are 12 of them in 2013 query set. From this, we conclude that the intersection approach worked well for 2/3 of the queries and poorly for 1/3 of the queries in the query set.

4.2 Conclusions and Suggestions for Future Research

In these experiments, we have attempted to gain insight into a difficult problem. Many times, in a research environment, we are given a reference run associated with a query set. This run specifies the top-ranked *n* documents retrieved by each of the queries. It is normally obtained by means of a carefully calibrated retrieval run. In general, much specific information must be present before such a run can be made.

In this case, we turn the problem on end and ask, given a query, is it possible to ascertain which documents of a very large corpus may be "of interest" with respect to this query? In this context, "of interest" means "related in some way" or "somehow connected"—not necessarily relevant, but potentially relevant. We would like to be able to identify this set of documents without first retrieving them.

There may be many ways to do this. One involves using the content of the documents, i.e., examining the text or portions of the text associated with the document before looking at the document as a whole. In these experiments, we have chosen this approach.

Results of these experiments are not definitive. Clearly, using intersection as the source set of documents is highly useful, but there is no guarantee that all documents of interest will be identified in this way. Intuition tells us that really good syntactic (statistical?) phrases related to or composed from query terms should be good discriminators in this context. If so, the phrases we used as filters clearly are not good enough. It seems that automatically generated phrases must be evaluated carefully before they can be used as we have used them here. We must be able to distinguish between very good phrases for this context and all the rest.

Many other context-based experiments may yield insight into this question. It is also reasonable to look at other aspects of the documents in question. The most promising of these may well be structure.

References

[1] About INEX [Internet]. Amsterdam, Netherlands. INEX: c 2008-2013 [cited 2013 August 20]. Available from: https://inex.mmci.uni-saarland.de/about.html/

[2] Crouch C. Dynamic Element Retrieval in Structured Environment. In: *ACM Transactions on Information Systems*; 2006.24(4): p 437-454.

[3] Extensible Markup Language [Internet]. World Wide Web Consortium: c 1996-2003 [cited 2013 August 20]. Available from: http://www.w3.org/XML/

[4] INEX Wikipedia collection and 2013 reference run [Internet]. Amsterdam, Netherlands. INEX: c 2008-2013 [cited 2013 August 20]. Available from: https://inex.mmci.uni-saarland.de/data/documentcollection.jsp

[5] Khanna, S. Design and Implementation of a Flexible Retrieval System [thesis]. Department of Computer Science, University of Minnesota Duluth; 2008.

[6] MontyLingua - A Free, Commonsense-Enriched Natural Language understander for English [Internet]. Boston, MA. MIT Media Labs: c 2002-2013 [cited 2013 August 20]. Available from: http://web.media.mit.edu/~hugo/montylingua/

[7] Salton G. *The Smart Retrieval System-Experiments in Automatic Document Processing*. Prentice-Hall; 1971.

[8] Salton G, McGill M. *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, Inc.; 1986.

[9] Salton G, Wong A, Yang C.S. A Vector Space Model for Automatic Indexing In: *Communications of the ACM*; 1975.18(11): p 613-620.

[10] Singhal A, Buckley C, Mitra M. Pivoted Document Length Normalization. In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*; 1996; New York. p. 21-29.