

Re-ranking Search Results based on Perturbation of Concept-Association Graphs

Gaurav Chandalia
CEDAR/Department of Computer Science and
Engineering
State University of New York at Buffalo
gsc4@buffalo.edu

Rohini Srihari
CEDAR/Department of Computer Science and
Engineering
State University of New York at Buffalo
rohini@cedar.buffalo.edu

ABSTRACT

There are many difficult IR queries that warrant the use of a more sophisticated content representation of a document corpus as opposed to the standard *bag of words* model. Moreover, effective algorithms are required to fully exploit the semantic information contained in such a representation. In this paper, we propose a characterization of the document corpus that is able to capture such information based on inherent relationships between concepts across the corpus. These relationships are then used to construct a concept-association graph from a small set of relevant documents with respect to the query. We also propose an algorithm based on perturbation of this graph and apply it to the task of re-ranking search results. Our technique, which is completely independent of the query employs a variation of the Subspace HITS algorithm [14]. Experiments carried out on the search results retrieved in the baseline run of the TREC 2003 HARD track show a significant improvement in precision as compared to the baseline run.

Categories and Subject Descriptors

H.4 [Information Storage and Retrieval]: Content analysis and indexing

General Terms

Algorithms, Experimentation

Keywords

re-ranking search results, matrix perturbation, link analysis, information extraction

1. INTRODUCTION

There are many IR queries which are intrinsically difficult, requiring either user feedback or advanced query processing techniques in order to obtain satisfactory results. An example of such a task is the HARD track. The goal of the High

Accuracy Retrieval from Documents (HARD) track of the *Text Retrieval Conference (TREC)* is to make the user part of the retrieval process and evaluation [2]. There are three phases in this track: baseline, clarifying and final. The results from the baseline phase are identical to those obtained from a classic TREC topic comprising of the title, description and narrative fields. The clarifying and final phases incorporate manual feedback from the user to construct a better and more accurate ranked list of documents. The clarifying phase was introduced to come up with more robust techniques to retrieve meaningful results for the intrinsically complicated queries. The purpose of the clarifying phase is to leverage more information from the user that would help to shed more light on the query or the kind of results preferred by the user. In the third phase, all the query and user information acquired in the second phase is used to improve performance. In other words, the last two phases combine to form a re-ranking technique - which is the main focus of this paper.

In this paper we address the issue of obtaining a more relevant list by processing the search results using an alternate representation of a document collection. The method that we propose is based on blind feedback as opposed to the manual feedback approach followed in the HARD track. And an alternate index comprising of concepts(terms) and associations rather than the standard IR index is used. The motivation behind our approach is that we need to capture important information about the context of a concept by looking at its links with other concepts. The representation of a document corpus as a standard IR index is not enough.¹ This standard IR index looks at the documents as *bag of words*. This is a simplification because we know that occurrence of index terms in a document or a document corpus are not uncorrelated. There are several techniques like Latent Semantic Indexing [8] that extract and represent the similarity of meaning of words and passages. In this paper however, we take a different approach - that of representing the document corpus as a set of concepts and associations that will help us distinguish between documents that talk about same concepts but different relationships.

The rest of the paper is organized as follows: Section 2 outlines prior work on re-ranking techniques and provides some references to work where the document corpus/document has been given a treatment similar to our concept-association representation. Section 3 gives an overview of our approach. The details of the concept-association index and its motiva-

¹The standard IR index is based on the *tf-idf* model [4].

tion are given in section 4. We briefly describe two link based techniques and present our algorithm in section 5. Experimental results are presented and analyzed in section 6. We conclude and discuss some directions of future work in Section 7.

2. RELATED WORK

Even though a lot of work has been done in re-ranking search results, it still remains an open problem with room for improvement. Re-ranking is popular as opposed to the traditional tasks of text based ranking which suffers from several drawbacks like polysemy, synonymy and short queries to name a few [4]. There are two sub-directions in this field - query dependent re-ranking and re-ranking by post processing of the search results. In query dependent ranking, initially the query is used to extract relevant pages from the collection using text based ranking methods. The query based methods modify (expand) the query using different techniques such as analyzing the corpus to discover word relationships or by analyzing the documents retrieved by the initial query [17]. The post processing techniques on the other hand, do not modify the query. However, they may or may not be independent of the query. One such approach clusters the search results and re-ranks the documents based on these clusters [10].

Here we are focused on post-processing search results from a single ranking system. We have explored link based approaches to present a new algorithm as a post processing method for re-ranking search results. Popular link based approaches like HITS [11] and PageRank [6] can be used to analyze the link structure of the pages/documents in the collection. Such methods are based on the fact that these links carry meaningful information which can be used to identify the relevance of pages to the query. As a precursor, the reader is referred to the following citations that have identified several problems with the HITS algorithm. [12] suggests a slightly modified version of the adjacency matrix constructed in the HITS algorithm. On the same lines, [5] improves upon the HITS algorithm by incorporating content analysis. On the other hand, [13] analyzes the stability of the PageRank and HITS algorithms using matrix perturbation on the basis of which two new algorithms are presented in [14]. One of these new algorithms - Subspace HITS, is based on the fact that although eigenvectors are very sensitive even to small perturbations, the subspace spanned by them is not.

There are several techniques that look at the semantic content of documents. The idea of extracting concepts and associations to extract features from the documents has been used widely for query expansion and building thesauri [7]. However, to our knowledge the idea of performing link analysis as a post processing step on such a content representation to re-rank search results has not been explored before.

3. OVERVIEW OF APPROACH

Before we get into the details of our technique, a high level overview of our approach is presented next.

Figure 1 shows the modules involved in re-ranking search results. The method can be divided into two modules - one module generates the Concept-Association graph and the second module is basically the algorithm that is used to re-rank the search results. The process begins with ob-

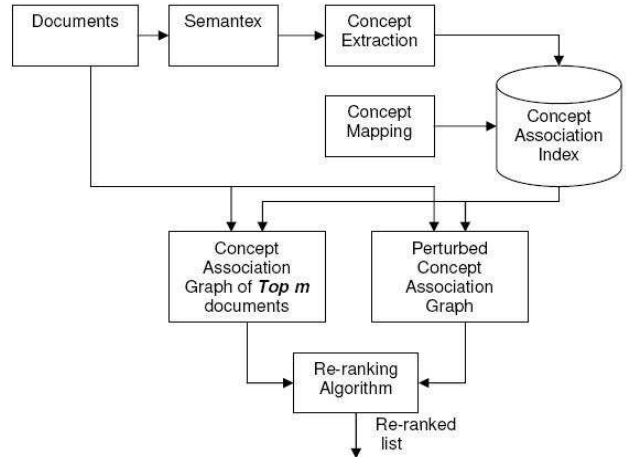


Figure 1: Steps in Generating Concept-Association Graph and Re-ranking documents

taining the documents searched for a given query using the standard IR index. These documents are then processed using an Information Extraction system to generate concepts and associations across the whole set. The resulting concepts and associations that are stored as an index are further used to construct a concept-association graph. The generation of the concept-association graph is explained in the next section. This graph however, consists of concepts and associations only from the *top m* documents. This is input to the algorithm along with a perturbed version of the graph that takes into account one document at a time from the remaining set of documents. The output of the algorithm is the re-ranked list for that query.

4. CONCEPT-ASSOCIATION INDEX

This section describes how we generate *concepts* and *associations* as well as the construction of the concept-association index. We obtained a beta-release version of the Semantex product [15] to analyze documents and extract candidate concepts and associations. The resulting set of extracted concepts is used as the indexing vocabulary for the document collection; this process significantly prunes this set. These concepts and associations are used to construct an adjacency matrix which in turn is used for reranking the document list. We used the following information produced by the engine to construct the index:

- **entities:** Named Entity objects represent key items such as names of *person*, *organization*, contact information such as *address*, *email*, *phone number*, *URL*, time and numerical expressions such as *date*, *year*, *etc.*
- **relationships:** relationships capture relationships between entities such as the *affiliation* associated with a person, for example, *Tom Smith, head of Purchasing at Softsoft Inc.*
- **events:** General Events are subject-verb-object-complement (SVOC) patterns representing 'who did what to whom when and where' at the logical level.

The following strategy is adopted in extracting concepts. First, all named entities (and their aliases reflected in the

Entity Profile) are considered to be concepts. Second, arguments of general events, specifically, the subject, verb and object slots are considered to be key concepts; many of these arguments are noun groups such as *air safety*. We perform normalization of words, rather than the traditional stemming used by IR systems; the root form of the word is returned. Furthermore, phrasal verbs such as *step down* and *take off* are currently recognized and grouped.

With respect to extracting associations, we employ a strategy of using both *labeled* and *generic* relationships between concepts. This strategy is required since otherwise many significant relationships would be missed. All labeled relationships (including syntactic relationships) are used. Since general events associate concepts through verbs, these are used as generic relationships. This is an alternative to using sentence level co-occurrence as evidence of an association since it reduces spurious associations.

In terms of a layered architecture, the standard IR index (concept-document) forms the bottom layer of our representation and facilitates key word and phrase searches. The Concept-Association index has additional layers built on top of the IR index to facilitate efficient search for concepts and associations. In our case, this index is used for producing a re-ranked list of documents. The original ranking is obtained by querying the standard IR index. It is noted that the index can store multiple associations extracted between two concepts.

We now describe a graph-based representation of the concepts and associations extracted from a document collection. Consider a graph labeled $G = (V, E)$ where the nodes are the concepts and the edges are the associations between these concepts. This representation is similar to the representation of the *World Wide Web* in the HITS algorithm. Instead of the pages as nodes, we have the concepts as nodes. We describe how this index is used for re-ranking in the next section.

5. LINK ANALYSIS METHODS

We have done richer content analysis of the corpus by exploring link based approaches to the above graph representation. Before presenting our algorithm, we briefly explain the HITS [11] and the Subspace HITS [14] algorithms - the bases of our algorithm.

5.1 HITS

The hypertext-induced selection (HITS) algorithm [11] is an effective technique for identifying *authoritative* sources in a hyperlinked environment such as the *World Wide Web*. In the hyperlinked environment, *authorities* are the ones that have a high number of *hub* pages pointing to them and *hubs* are the ones that point to a high number of *authoritative* pages. Thus in the HITS algorithm, a graph G is constructed with the pages as the nodes and links between the pages as the edges between the nodes. More formally, we can represent this graph as a square adjacency matrix A of n dimensions wherein an entry between two pages is 1 if there is a link between them and 0 otherwise. The idea is to associate authority and hub weights with each page and update them iteratively on the basis of their definitions. The authority weight of a page depends on how many pages having large hub weights point to that page and *vice versa* for the hub weights.

We know from theorem 3.1 in [11] that the authority and

hub values a_i and h_i converge to the limits a^* and h^* respectively. Also, from theorem 3.2 in [11] it follows that a^* is the principal eigenvector of $A^T A$ and h^* is the principal eigenvector of AA^T and a_i^* and h_i^* respectively indicate the extent to which the page i is an authority and a hub.

5.2 Subspace HITS

In [13], it was shown that the HITS algorithm is not stable to small perturbations. To be more precise, it was observed that there is a substantial fluctuation in the page rankings when a small number of links were added/deleted from the graph. This laid the foundation of the Subspace HITS algorithm which computes the authority values by projecting each eigenvector on the subspace spanned by the top k (where $k = n$ in the ideal case) eigenvectors. The reason is that even though the eigenvectors are sensitive to perturbations, the subspace that spans these eigenvectors is not [16]. The Subspace HITS algorithm is very similar to the HITS algorithm except for one important difference - the way in which the weights are computed. This algorithm considers the eigenvectors of the matrix $A^T A$ as the basis vectors for a subspace and then uses the projection of each basis vector onto the subspace to compute the authoritative values. We present a variation of this algorithm in the next section where we discuss our algorithm in detail. But for now, it suffices to say that this algorithm is more stable than the HITS algorithm as it does not try to interpret the individual eigenvectors, but considers the effect on the whole subspace. This fact is further reinstated by our first set of results presented in section 6 where we tried to interpret the individual eigenvectors as topics within a document.

5.3 Perturbed Subspace HITS for Re-ranking

This section explains our algorithm in detail and the steps involved in the process of re-ranking are elaborated.

Let's say that we have a concept-association index and a ranked list of documents that we want to re-rank by taking advantage of the semantic information that we have encoded in the index. Here, we make an assumption that the top m documents in the ranked list are relevant to the query. In the real world, we know that this is always not true, however making the assumption enables us to concentrate on the main problem of re-ranking documents. We construct a *graph* as explained in Section 4 by considering concepts and associations only from the top m documents.² Now consider a scenario wherein we perturb the graph. The perturbed graph can still be represented as being relevant to the query if the change in the graph is ideally none or minimal. If there is no change, then it implies that the perturbation that was added can be characterized as being relevant to the complete original graph or some portion of it and hence the query, and *vice versa*. Another interpretation is that the perturbation was not enough to bring about any change in the graph.

The above scenario is realistic and will hold true if we know two things - how to measure the change in the graph and the nature of perturbation. We first consider the nature of perturbation - it has to be in accordance with the graph's representation to facilitate comparison of the perturbed graph and the original graph. Let's suppose that we perturb the original graph by adding associations between certain concepts. Moreover, we add only those associations

²The graph here is *symmetric*.

that have been found in the $(m + 1)^{th}$ document. Also, the set of concepts that are considered for perturbation are an intersection between the concepts of the top m documents and $(m + 1)^{th}$ document. Note that the perturbation, although symmetric, is not random. We employ a variation of the Subspace HITS algorithm to measure the change in the graph. If we consider the graph to be represented by an adjacency matrix A , then we can compute the differences in the authority scores of the perturbed and original representations. For computing the differences, ideally we should consider all the eigenvectors ($k =$ dimension of the original graph). However, for computational reasons we choose $k = 25$ which means that we do not consider the eigenvectors that have small weights.

The process starts with processing the whole document collection to extract the concepts and associations between concepts. Consider a query q and a ranked list of n documents retrieved based on the standard IR index for this query. Let \mathbf{doc} be a document. Construct a concept-concept square adjacency matrix A by considering concepts and associations only from the top m ranked documents. This matrix will be the primary graph that represents the query q .

The algorithm is as follows:

1. Compute the top k eigenvectors $V^a = (\mathbf{v}_1^a, \mathbf{v}_2^a, \dots, \mathbf{v}_k^a)$ and the corresponding eigenvalues $d_1^a, d_2^a, \dots, d_k^a$ of $T = A^T A$.³
2. Compute the authority scores for $T = A^T A$ as $\mathbf{x}_s^a = \sum_{p=1}^k d_p^a ((\mathbf{v}_s^a)^T \mathbf{v}_p^a)^2$ where $s = 1$ to k .
3. Loop for $i := (m+1)$ to n .
 - (a) Initialize $B = A$. Perturb B using the concepts and associations from \mathbf{doc}_i .
 - (b) Compute the top k eigenvectors $V^b = (\mathbf{v}_1^b, \mathbf{v}_2^b, \dots, \mathbf{v}_k^b)$ and corresponding eigenvalues $d_1^b, d_2^b, \dots, d_k^b$ of $S = B^T B$.
 - (c) Compute a vector of change in authority scores for \mathbf{doc}_i with respect to the top m documents in the following way:
$$\mathbf{f}_s = \sum_{p=1}^k d_p^b ((\mathbf{v}_s^b)^T \mathbf{v}_p^a)^2 - \mathbf{x}_s^a; \quad s = 1 \text{ to } k$$
4. EndLoop
5. Re-rank the documents using *Borda count* by taking each eigenvector as the *voter* and the documents from $(m+1)$ to n as the *candidates*.

Note that in step 2, we compute the projection of each eigen vector onto the subspace spanned by the set V^a . Thus, the vector \mathbf{x}^a is nothing but the vector of eigenvalues of T . This step is not required, however we have shown this to be consistent with step 3c where we do similar computations. The *sum* in the step 3c computes the square of the length of the projection of an eigenvector of V^b on the subspace spanned by the basis set V^a . The term d_p^b is the weight we assign to give more importance to the eigenvectors with high eigenvalues.

³ \mathbf{d} is a vector of eigenvalues and in the case of repeated eigenvalues, we assume the eigenvectors are chosen orthogonal to each other.

Borda Count is a simple, yet very effective technique that has been widely applied in the field of social sciences as a voting procedure. For n candidates, every voter submits his ranking of candidates: n points are given to the candidate at rank 1; $n - 1$ points to the candidate at rank 2 and so on. For each candidate, the points obtained from each voter are added and if the candidates are arranged in a descending order of the total points, then the one with the maximum number of points wins. This process is very useful because it gives a complete list of ranking of all the candidates, and for this reason it has been successfully applied for combining the predictions from different classifiers in pattern recognition [9] and also for combining the ranked lists of documents returned by multiple search engines[3].

6. EXPERIMENTS & ANALYSIS

The aim of this research is to push the relevant documents higher up in the ranked list of documents. Hence, we wanted our dataset to have the following characteristics: have a substantial number of relevant documents retrieved for a query, and quite a few of the retrieved relevant documents occur in lower ranks. This is the reason we chose the 2003 HARD track dataset. The HARD corpus consists of newswire text from the 1999 portion of the ACQUAINT corpus (New York Times, Associated Press Wordstream, Xinghua English) and of U.S. government documents (Federal Register, Congressional Record) [2]:

	NYT	APW	XIE	CR	FR	Totals
No. of docs.	137,806	77,876	104,698	16,609	35,230	372,219
Size	750MB	245MB	310MB	147MB	330MB	1.7GB

Table 1: Data used for the experiment.

The 2003 HARD track contained 50 queries. We eliminated the queries for which the baseline run retrieved less than m relevant documents as the task of re-ranking would not be applicable in that case because of absence of relevant documents in the remaining set of documents retrieved for that query. Out of the remaining set, we randomly picked

Query No.	Query
33	Animal Protection
48	Y2K crisis
51	Hate Crimes Prevention
65	Mad Cow Disease
69	Environmental Protection
70	Red Cross activities
77	Insect-borne illnesses
84	Recent Earthquakes
99	Globalization and Democracy
102	Microsoft monopoly
116	Genetic Modification technology
146	NATO/UN Tension over Balkans crisis
147	Regional Economic integration
187	National Leadership Transactions

Table 2: Queries evaluated in the experiment.

a subset of queries. Table 2 shows the queries that we have evaluated and table 3 gives details of the number of relevant

documents in the query and the number of relevant documents that were retrieved in the baseline run. We have used the standard *TREC evaluation code* [1] to evaluate the results. For our experiments, we have considered the first 500 documents retrieved for each query and the value of m was set to **10**. More discussion on setting the value of m follows in the next section.

Query No.	Relevant Docs. in the collection	Docs. Retrieved in <i>baseline</i> run
33	401	211
48	562	301
51	168	140
65	145	126
69	513	101
70	111	90
77	194	25
84	86	43
99	399	170
102	285	249
116	200	116
146	305	131
147	327	181
187	194	51

Table 3: Number of relevant docs. present in the collection and number of relevant docs. that were retrieved for each query.

Before presenting the results, some analysis of the algorithm is presented here. It was stated in Section 5.3 that the adjacency matrix is perturbed by taking into account concepts and associations from a single document at a time (for e.g. 11th document). Now, consider the original graph again. The graph will be a single connected component if the query essentially represents only one topic. However, as is observed in real world datasets, the graph consists of multiple connected components thus indicating that the query can be representative of more than one distinct topic or rather in our case, there could be concepts that are related to more than one topic (aspect) of the query. Also, we refrain from interpreting individual eigenvectors as *individual topics*.⁴ It is very likely that each eigenvector is probably a combination of different authoritative nodes. In fact, this is the reason why we want to consider the graph of *top m* documents.⁵ We are trying to make use of the fact that the top k eigenvectors are a linear combination of different authoritative nodes that have associations across these m documents. These authoritative nodes could be from different connected components. However, at this point we should mention that we are not trying to interpret the different connected components as representing distinct topics. The idea is to simply understand the structure of the graph of the *top m* relevant documents and use this information to find out whether other documents are relevant to the query or not. The hypothesis is that if we can measure the change that occurs due to disturbing the structure in the graph in a methodical manner, then we can obtain

⁴This can easily be proved by considering a small 10 by 10 matrix and then looking at the effect of perturbing certain highly connected/isolated nodes.

⁵It is not a hard and fast rule that we consider only 10 documents; certainly this number can be experimented with to improve the performance.

information about the relevance of the disturbance with respect to the query. Even though eigenvectors are sensitive to perturbation, since we have considered the projection of the perturbed eigenvectors on the subspace, the algorithm is more stable to perturbation. Moreover, in our case it makes sense to consider the subspace formed from eigenvectors as basis because this eliminates the need of interpreting the individual eigenvectors under perturbation; rather consider the effect of perturbation on the whole subspace.

Consider a new document from which we select some concepts. Depending on the structure of the graph, when some associations are flipped among these concepts there are several cases that have to be looked at:

- if concept a belongs to one connected component and concept b belongs to another component; and if previously no association exists between a and b , then introducing an association between them will result in some noise. If there are several such situations, then a considerable amount of noise will be introduced in the perturbed graph. Hence even though the new document has some concepts in common with the original graph, we can say that it is not relevant because perturbing the graph with these associations results in more noise.
- if the concepts a and b are present in the same connected component; if previously *no association exists* between a and b , then introducing an association between them will not result in much variation. Thus even if there are more such concepts, it implies that these concepts are relevant as they retain more or less the structure of the original graph. Hence, this document will have a high relevance.
- consider the case when concepts a and b already *have an association* in the original graph. If there are more such concepts then there will hardly be any change between the perturbed and the original graphs. This also implies that the document being considered is relevant.
- there is a problem however if the concepts from the new document occur in isolation and do not belong to any of the connected components in the original graph. In this case, even if we perturb their associations, the top eigenvectors will not be affected much. But this does not mean that the document is relevant. However, another interpretation is that such concepts occurring in isolation probably do not play a vital role in contributing to the relevance of the document to the query. Thus the information that we have from the *top m* documents is not enough to say more about the relevance of this document by considering such concepts. This is a case in which an irrelevant document is pushed higher in the rank.
- Another case whose outcome is same as that of the previous case is when there are no concepts that are common to the new document and the original graph.

The last two cases are instances in which the algorithm will place irrelevant documents higher in the rank. Such instances will effect the precision only when they are large in number i.e. there are many documents that have concepts that occur as isolated nodes in the original graph.

We have described the ideal scenarios above; there can be several others but they can be explained as one or more variations/combinations of the ideal scenarios.

Next we present the first set of results where we applied the HITS algorithm on the dataset. The main algorithm from Section 5.3 remains the same with the Subspace HITS step replaced by HITS algorithm. Instead of computing the authority scores, we computed the cosine similarity value between the corresponding eigenvectors. Figure 2 shows the precision over the queries from Table 2. The dashed line represents the results from the 2003 baseline run and the solid line represents the evaluation of the queries using the current algorithm. There were a lot of fluctuations in the eigenvectors on account of perturbation and hence it was not surprising that there were considerable differences in the similarity values. Subsequently, although the graph shows that the algorithm performed better for two queries (*Hate Crime Prevention* and *Globalization and Economy*), there is no real basis why some of the documents were pushed higher up the order based on the original graph. Also, the average precision over the 14 queries actually decreased by 16 percent as compared to the baseline run. This shows that in our case of concept-concept graph representation, it is probably not a good idea to interpret individual eigenvectors. Rather, consider the subspace spanned by the eigenvectors which as our next set of results show, seems to perform better than the current algorithm.

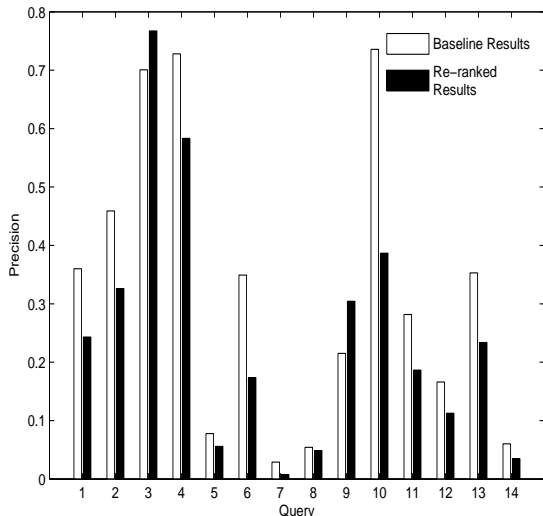


Figure 2: Performance of HITS algorithm. Exact precision of each of the 14 queries.

Figure 3 shows the precision of each of the queries when we ran the algorithm from Section 5.3 as stated. There was an increase in precision for 10 of the 14 queries. Out of these 10 queries, 7 queries showed an improvement of at least 20 percent. Moreover, for 13 out of the 14 queries, the precision at 200 documents improved with 8 of them showing an improvement of over 15 percent. Figure 4 shows the precision at 200 documents retrieved for the 14 queries. These results are consistent with our aim of pushing the relevant documents from very low ranks to the top. However, out of the 4 queries that obtained lower precision than their baseline

counterparts, 2 queries (*Insect-borne illnesses* and *Recent Earthquakes*) showed a dip of more than 35 percent. We believe that the cause for this is the assumption that we make in the beginning - only the *top m* documents are relevant to start with.

Table 4 shows the results of our algorithm for the query *Y2K Crisis*. The first column represents the name of the relevant document, the second column corresponds to the rank of the document obtained from search on the standard IR index and the third column corresponds to the rank of the document obtained from applying the Perturbed Subspace HITS algorithm.

The query *Recent Earthquakes* showed the highest dip in precision as compared to the 2003 baseline results. Table 5 shows some of the relevant documents that were pushed lower down the rank. As we mentioned before, that our algorithm's performance is highly dependent on the assumption that the *top m* documents are relevant. For this query's baseline results, the precision at 10 documents retrieved was 0.0. And for this reason, this is the only query for which very few relevant documents were pushed up in the ranked list. At the same time, because of the irrelevance of the top documents with respect to the query some of the relevant documents ranked between 10 to 100 were pushed further down resulting in lower precision than that of the baseline run.

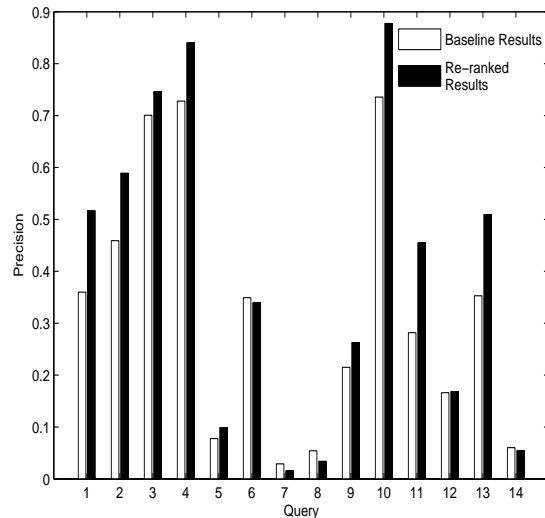


Figure 3: Performance of our proposed Perturbed Subspace HITS algorithm. Exact precision of each of the 14 queries.

7. DISCUSSION & FUTURE WORK

We have presented a fairly robust technique for re-ranking search results that is not explicitly dependent on the query. Experimental results show a substantial improvement in precision indicating that the technique is quite effective. The method exploits the information contained in the associations among important concepts. Our method is robust because it shies away from interpreting the individual eigenvectors as topics; rather it looks at the perturbation effects

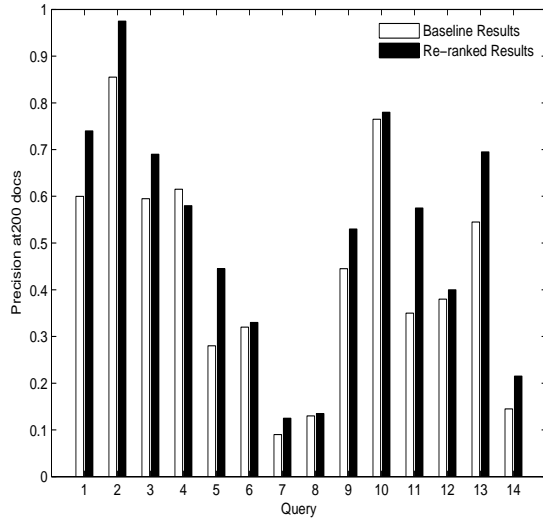


Figure 4: Performance of our proposed Perturbed Subspace HITS algorithm. Exact precision at 200 documents retrieved of each of the 14 queries.

APW19990119.0237	133	53	NYT19990315.0243	136	49
XIE19990412.0107	171	25	NYT19990119.0104	173	75
NYT19990418.0180	205	42	APW19991006.0056	208	66
NYT19990305.0149	211	59	XIE19991203.0191	224	26
NYT19991112.0025	225	18	NYT19990104.0187	229	94
XIE19990422.0199	231	164	NYT19991220.0074	236	146
APW19991021.0320	239	30	XIE19991229.0117	242	17
NYT19990909.0430	243	74	NYT19990111.0075	245	123
XIE19990305.0009	247	64	CRS19990928.0020	248	44
NYT19990321.0160	254	159	NYT19991026.0207	260	21
NYT19991209.0059	265	43	APW19990608.0013	279	248
XIE19990721.0071	282	216	NYT19990327.0179	283	192
XIE19991107.0078	284	262	NYT19990326.0333	286	80
NYT19991222.0214	289	189	XIE19990304.0157	291	256
XIE19990422.0395	294	98	APW19990111.0142	296	102
XIE19990412.0231	298	227	NYT19991215.0072	299	148
NYT19991210.0094	301	129	NYT19991215.0056	303	160
NYT19990716.0329	304	214	XIE19991227.0022	307	193
NYT19990625.0396	308	204	XIE19990304.0158	329	11
NYT19991230.0010	333	110	NYT19990224.0231	334	145
APW19990930.0194	337	117	NYT19990303.0108	339	56
NYT19990419.0076	341	68	NYT19990410.0017	343	279
XIE19990423.0070	344	186	XIE19991001.0211	347	108
NYT19990706.0070	351	166	XIE19991109.0225	352	168
XIE19991101.0158	353	29	NYT19991030.0045	355	191
NYT19991230.0049	356	194	NYT19991025.0405	357	124
NYT19991230.0009	363	275	NYT19991229.0377	364	126
APW19990922.0021	365	128	NYT19991025.0192	366	268
NYT19991230.0054	368	257	APW19990331.0020	369	318
NYT19990818.0093	371	38	NYT19991021.0273	375	260
NYT19991021.0272	376	170	NYT19990215.0064	377	174
CRS19990518.0005	379	141	NYT19990304.0132	382	322
APW19990605.0128	387	134	NYT19990722.0193	390	58
XIE19990211.0204	393	234	APW19990302.0180	395	113
NYT19991020.0290	397	57	XIE19990125.0117	398	208
XIE19990904.0275	399	180	XIE19990904.0149	400	120
NYT19990517.0054	402	163	NYT19991219.0147	403	277
XIE19990419.0199	404	90	CRS19990518.0007	410	50
APW19990112.0162	411	314	XIE19991210.0205	416	242
XIE19991118.0060	417	63	APW19990723.0205	418	235
XIE19991227.0049	421	244	NYT19991021.0347	423	281
NYT19990714.0421	425	65	XIE19990915.0058	427	261
NYT19991217.0041	435	22	XIE19991211.0007	436	265
XIE19991220.0229	438	51	XIE19990130.0146	439	136
NYT19991118.0299	440	92	XIE19991221.0205	441	205
NYT19991116.0072	447	172	XIE19990920.0119	448	175
XIE19991221.0139	449	96	NYT19990724.0009	451	176
NYT19991213.0110	455	202	NYT19991231.0209	457	79
NYT19991229.0034	458	254	NYT19990522.0203	462	82
XIE19990611.0270	464	274	NYT19990606.0048	465	32
XIE19991123.0279	466	70	XIE19990422.0257	467	34
APW19990409.0200	470	217	XIE19990606.0010	484	23
NYT19990524.0272	485	13	NYT19990524.0143	489	203
NYT19990425.0163	492	151	NYT19990202.0257	494	111
XIE19991214.0204	496	206	XIE19990908.0224	497	33
NYT19990318.0070	498	69	NYT19990104.0182	499	232

Table 4: Relevant Documents related to query Y2K Crisis and their ranks before and after re-ranking.

CRE19990429.0032	18	485	NYT19990817.0349	19	322
NYT19990823.0241	25	425	NYT19990819.0127	26	482
NYT19990827.0263	27	237	NYT19990201.0084	28	419
NYT19990201.0083	29	165	NYT19990131.0009	30	167
NYT19990131.0008	31	169	XIE19990811.0093	32	173
NYT19990818.0024	34	284	NYT19990818.0023	35	381
NYT19991015.0200	36	387	NYT19990824.0272	39	331
APW19990907.0243	40	323	NYT19991027.0147	44	481
NYT19991027.0145	45	254	CRH19990421.0020	46	257
NYT19990922.0239	50	360	NYT19990510.0222	51	87
NYT19990510.0221	52	53	NYT19990823.0086	57	487
NYT19991011.0243	58	488	NYT19990205.0048	59	433
XIE19990119.0113	60	311	NYT19990821.0091	65	491
XIE19991125.0073	66	493	XIE19990626.0196	68	141
NYT19990912.0096	69	100	APW19991018.0237	72	125
APW19991026.0028	74	393	APW19990330.0203	75	398
APW19990330.0177	77	368	XIE19991017.0005	78	370
XIE19990317.0245	79	374	CRH19990209.0019	80	301

Table 5: Relevant Documents related to query Recent Earthquakes and their ranks before and after re-ranking.

on the graph represented by a small but relevant set of documents. At this point we should mention that the stability bounds presented in [14] will not affect the performance of the algorithm. The phenomenon of high perturbation resulting in an unstable graph is indicated by large differences in the authority scores of the original and the perturbed graph. This is implicitly captured by *Step 3c* of the Perturbed Subspace HITS algorithm. Subsequently, that particular document will be given a low relevance score.⁶ Thus the algorithm is more resistant to documents that have a lot of concepts but not too many associations in common to the original graph.

There are however two factors that decide the performance of this approach. The assumption that only the *top m* documents are relevant to start with is a deciding factor for the performance of the technique. This idea however is reasonable and widely observed, especially for techniques involving blind feedback. However, like the clarifying phase of the HARD track, we are looking at other options to build this initial set of relevant documents to avoid the problems that we faced for queries like *Insect-borne Illnesses* and *Recent Earthquakes*.

The current approach also relies on the structure of the original graph. A highly disconnected graph can certainly result in a lower precision than that of the baseline run.⁷ This can easily be compensated by varying the number *m* to obtain more associations. This suggests an interesting connection between the number *m* and the connectedness of the original graph to improve precision. Although the focus of this paper has been on exploiting a better content representation and proposing a novel re-ranking technique, exploring the connection between *m* and the connectedness of the graph is something that we are currently looking at. The other direction that we intend to focus on is improving the concept and association extractor as its performance directly influences the performance of our algorithm.

8. REFERENCES

- [1] Trec evaluation program.
<http://trec.nist.gov/results.html>.

⁶If the perturbed graph is unstable, then it is unlikely that the document that causes it is relevant to the query.

⁷A highly disconnected graph also means that there are a large number of connected components implying that the documents do not contain much semantic information that can be exploited.

- [2] J. Allan. High accuracy retrieval from documents - HARD track overview in trec 2003. In *NIST Special Publication 500-255: The Twelfth Text REtrieval Conference (TREC 2003)*. NIST, 2003.
- [3] J. Aslam and M. Montague. Models for metasearch. *Proc. SIGIR 2001*, September 2001.
- [4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, New York, USA, 1999.
- [5] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, 1998.
- [6] S. Brin and L. Page. The anatomy of a large scale hypertextual search engine. *17th International World Wide Web Conference*, 1998.
- [7] Z. Chen, S. Liu, W. Liu, G. Pu, and W. Ma. Building a web thesaurus from web link structure. In *Proc. of the 26th annual international ACM SIGIR, Canada, 2003*, pages 48–55, 2003.
- [8] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [9] M. V. Erp and L. Schomaker. Variants of the borda count method for combining ranked classifier hypotheses. In *In Proc. 7th International Workshop on Frontiers in Handwriting Recognition*, pages 443–452, September 2000.
- [10] M. A. Hearst and J. O. Pedersen. Re-examining the cluster hypothesis: scatter/gather on retrieval results. *In Proc. 19th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 76–84, August 1996.
- [11] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *In Proc. 9th Ann. ACM SIAM Symp. Discrete Algorithms*, pages 668–677. ACM, 1998.
- [12] J. Miller, G. Rae, and F. Schaefer. Modification of Kleinberg’s HITS algorithm using matrix exponentiation and web log records. *In Proc. SIGIR 2001*, pages 444–445, September 2001.
- [13] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Link analysis, eigenvectors and stability. In *In Proc. 17th International Joint Conference on Artificial Intelligence*, 2001.
- [14] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *In Proc. 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.
- [15] R. K. Srihari, W. Li, C. Niu, and T. Cornell. Semantex: A customizable intermediate level information extraction engine. *To appear in the Journal of Natural Language Engineering*, 12(4), 2006.
- [16] G. W. Stewart and J. Guang Sun. *Matrix Perturbation Theory*. Academic Press, Inc., London, UK, 1990.
- [17] J. Xu and W. Croft. Query expansion using local and global document analysis. *In Proc. ACM SIGIR*, 1996.