Sentence Level Information Retrieval

A dissertation submitted in partial fulfillment of the requirements for the M.Tech (Computer Science) degree of the Indian Statistical Institute, Kolkata.

By

Swarup Chandra Sahoo

under the supervision of

Dr. Mandar Mitra
Computer Vision & Pattern Recognition Unit

INDIAN STATISTICAL INSTITUTE
203, B.T. Road, Kolkata - 700035
21 July, 2004

Indian Statistical Institute,
203, Barrackpore Trunk Road,
Kolkata - 700 035

Certificate Of Approval

This is to certify that this thesis titled " *Sentence Level Information Retrieval*" submitted by Swarup Chandra Sahoo towards partial fulfillment of requirements for the degree of M.Tech in Computer Science at Indian Statistical Institute, Kolkata embodies the work done under my supervision.

Mandar Mitra

Dr. Mandar Mitra
Computer Vision & Pattern Recognition Unit,
Indian Statistical Institute,
Kolkata - 700 108.

(External Expert)

Acknowledgement

I take pleasure in thanking Dr. Mandar Mitra for his friendly guidance throughout the dissertation period. His pleasant and encouraging words have always kept my spirits up.

Finally I take the opportunity to thank my classmates, friends and family members for their encouragement to finish this work.

Swarup Chandra Sahoo

Abstract

With the proliferation of sources of information on WWW and on storage devices, information retrieval technique have become an integral part of human life. In this study, we experiment on expansion based approaches. Earlier retrieval systems used to retrieve relevant documents. Large size of documents are not helpful in locating the relevant part within it. So, we need a system to retrieve only relevant portion of documents for the user. In this report, we focus on the retrieval of relevant portions of the documents.

Contents

Chapter 1

Introduction

1.1 Information Retrieval

Information Retrieval(IR) is concerned with locating information that will satisfy a user's information requirement. Traditionally, the emphasis has been on text retrieval: providing access to natural language texts where the set of documents to be searched is large and topically diverse. In other words, IR deals with the retrieval of unstructured information. Recently researchers have expressed interest in information retrieval from a collection regardless of the medium that happens to contain that information. Applications of IR include WWW searching, searching through CD-ROM encyclopedias or e-books, searching through specialized document collections on medicine, literature etc. These document collections may include non-text based information, i.e. images, audio, video etc. For the current dissertation, the discussion is restricted to text based IR.

In a typical application, a collection of documents may be several gigabytes in size and will contain on the order of millions of documents. In a collection of this size. often only a few hundred documents, or fewer, will be relevant to a specific query. This large disparity in the size of the relevant set of documents vs. the non-relevant set makes the problem quite different from many classification tasks and affects how retrieval systems are designed as well as how they are evaluated.

1.2 Definitions in IR

Before proceeding further. it is important to define several significant terms. A *query* is a text expression that describes the information need of the user. The query is passed to the *retrieval system*. which uses this information to determine which documents in its collection are *relevant* to the user's request. A *document* is the organizational unit of the information collection. The *collection* consists of a large number of documents. A *relevant document* is one which contains information related to the query.

A natural query can range from a simple phrase like "information superhighway" to a longer statement like: "references to cluster analysis in the context of the information retrieval" and to even longer passages that may consist of several pages of text. It is a common strategy in IR to incorporate the text of full documents into a query.

An IR system matches the query against the collection and returns a set of documents to the user, often ranked in order of their estimated relevance. The number of relevant documents retrieved provides an objective measure of system performance. This measure is usually normalized in one of two different factors, i.e., *precision* and *recall*. Precision is defined as the number of relevant documents retrieved over the total number retrieved. Recall is defined as the total number of relevant documents retrieved over the total number of relevant documents present in the collection.

1.3 Background

1.3.1 The Vector Space Model

The Vector Space Model is one of the most commonly used models in the field of IR. Under this model any given text is represented as a list of *terms* or *keywords* with associated *weights*. A term is usually a word or a phrase and the weight corresponding to a term is a measure of its importance in representing the information in the given text. If the total number of distinct terms in a collection of texts is T, then, in this model, a text D_i can be represented as a vector in a T-dimensional vector space:

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{iT})$$

where, d_{ik} is the weight of term t_k in document D_i. A weight is zero if a term is absent from a particular document, and positive weights characterize terms contained in a document.

1.3.2 Similarity

The relatedness of two pieces of text – a query and a document, for example – can be estimated by measuring the closeness of the corresponding vectors: when two vectors are "close", the corresponding texts are expected to be semantically related. In the vector space model, the closeness of two vectors can be measured using the vector inner product. The relatedness of document $D = (d_1, d_2, \ldots, d_T)$ and a user query

$Q = (q_1, q_2, \ldots, q_T)$, called the *similarity*, is thus given by

$$Sim(D, Q) = \sum_{i=1}^{T} d_i \times q_i$$

Documents from the collection can then be ranked in decreasing order of their similarity to the query. Highest ranked documents are expected to be the most useful for the user.

1.3.3 Indexing

Indexing is the method by which terms are assigned to a document and weights are computed for assigned terms.

Term Assignment: The list of terms assigned to a text is typically obtained using the following steps:

1. Tokenization: The text is first broken into individual words, punctuation marks, and other tokens.
2. Stopword removal: Common words (also called stopwords) like *the,of, an*, etc. are removed from the list of words obtained above.
3. Stemming: Morphological variants of a word are normalized to the *stem*. For example, "believing" is converted to "belief". Similarly, "believes" is also converted to "belief".
4. Phrase recognition: Multi-word phrases (e.g. "information retrieval", "computer science") are recognized and added to the list of single words to index the text.

Term Weights: The quality of document ranking is crucially dependent upon the assignment of proper weights to terms in the texts. Most modern IR systems assign weights to the terms in a text using the following three factors:

1. **Term Frequency** (*tf*) is the number of occurrences of a term within a document. Documents that repeatedly use a query term are potentially more useful than documents that rarely use that query term. Therefore, the weight of a term should be an increasing function of its *tf*.
2. **Inverse Document Frequency** (*idf*) is an inverse measure of the number of documents in the collection in which a term occurs. Words that are used in numerous different documents are less important than words that are used in a few documents. Thus, the weight of a term should be an inverse function of its document frequency.

3. **Document length:** Long documents often tend to repeat terms and thus, in general, have higher term frequencies. Long documents also use numerous different words. Thus the number of matches between a query and a long document tends to be high. For these reasons, long documents can get a preference in retrieval over short documents just because of their length. Therefore, document term weights should be scaled down using the length of the document. This is called *document length normalization.*

Using these factors, we can define the weight of a term in a document as:

$$\frac{tf \times idf}{document\ length}$$

1.4 Classical IR Problem

The task is to match the query against the document collection and return relevant documents to the user. The success of the system will depend significantly on the quality and quantity of the information associated with the query. If a lot of data is available about what defines a relevant document, then the system may be able to employ more advanced categorization techniques on the collection.

The optimal response to a query would be for the system to find all the relevant documents and return nothing that is not relevant. This will almost never happen in practice, as the retrieval task can rarely be executed with such accuracy. However, the more documents that are returned to the user, the larger the number of relevant documents that will be found. But as the system considers documents that are less and less likely to be relevant, the percentage of non-relevant documents found will begin to increase. Therefore, it is clear that there is a trade-off between finding more relevant documents(recall) and being forced to examine non-relevant documents(precision). The searcher must decide how many non-relevant documents he is willing to examine in order to discover an additional relevant document.

One of the primary measures used to evaluate retrieval results is the F measure, defined as

$$F = \frac{2 \times P \times R}{P + R}$$

where, P is precision and R is recall. The F measure can be used to compare the performance of different IR systems.

1.5 Sentence Level IR

Often, the information of interest to a user is contained in a small portion of the retrieved document, while the rest of the document is irrelevant. If the documents' average size is large, then we need to filter the retrieved documents one level more, so that only the relevant portions of retrieved documents can be generated as a result of the user query. Starting from the ranked list of retrieved documents, a system's task is to first filter out all non-relevant information from those documents, reducing it to the essential components of relevance-defined to be *sentences* that were relevant. As an extension, the system can scan those relevant sentences and discard any that do not contain new material, defined as *novel* sentences.

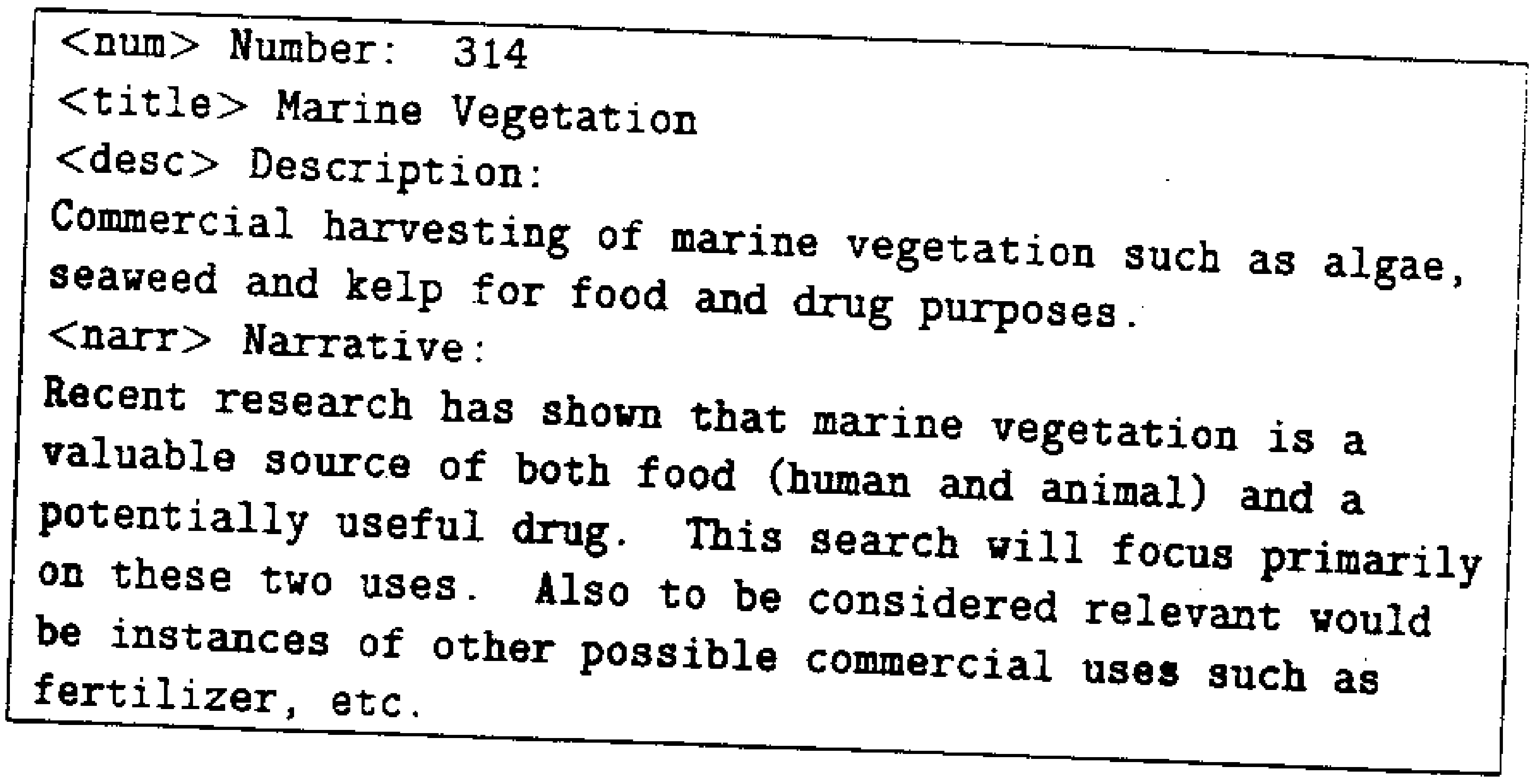
There are several reasons to tackle this task at sentence level. The initial intuition was that by reducing the granularity of text unit, it would be easier to identify novel or redundant information. An earlier suggestion was to consider a passage as a text unit. Because the size of a passage is not easily defined and paragraphs are not always available, so sentences were chosen as the text unit for retrieval. During sentence level retrieval, each document is split into sentences, and each sentence is considered as a small document. This new set of documents is used in the retrieval process.

Queries supplied by users are typically short and contain few key-terms. Expanding the query by adding terms related to the original query terms results in significantly improved performance. One widely used method for query expansion uses *relevance feedback*. In this method, the user provides feedback about a few top-ranked documents retrieved in response to the original user query by specifying which are relevant and which are not. The system uses this information to modify the query. Terms that have a significant presence in the relevant documents but don't occur much in the non-relevant ones are added to the query and final term weights are determined based on the occurrence patterns of the terms in the relevant and non-relevant documents. The modified query is then run against the database to retrieve the final list of documents.

Adhoc expansion, a completely automatic, "pseudo" relevance feedback method that does not require actual feedback from the user has also been found to yield substantial improvements. In this method, a small set of (say 20) documents is retrieved using the original user-query; these documents are all *assumed* to be relevant (without any intervention by the user) and used in the feedback process described above to construct the expanded query, which is then run again to retrieve the set of documents actually presented to the user.

1.6 TREC

The Text REtrieval Conferences (TREC) [1], co-sponsored by the National Institute of Standards and Technology (NIST) and U.S. Department of Defense, was started in 1992 with a purpose to support research within the information retrieval community



```
<num> Number:  314
<title> Marine Vegetation
<desc> Description:
Commercial harvesting of marine vegetation such as algae,
seaweed and kelp for food and drug purposes.
<narr> Narrative:
Recent research has shown that marine vegetation is a
valuable source of both food (human and animal) and a
potentially useful drug.  This search will focus primarily
on these two uses.  Also to be considered relevant would
be instances of other possible commercial uses such as
fertilizer, etc.
```

Figure 1.1: A sample TREC 2002 topic from the novelty track.

by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. For retrieval experiments, TREC provides test collections which are large enough so that they realistically model operational settings.

TREC calls a natural language statement of information need a *topic* to distinguish it from a *query*, which is the data structure actually presented to a retrieval system. The topics are formatted using a very simple SGML-style tagging. A topic statement generally consists of four sections: an identifier, a title, a description, and a narrative. An example topic taken from the TREC 2002 novelty track is shown in Figure 1.1.

TREC provides English document collection volumes in 5 disks and 450 topics for these documents. This collection includes materials from various news-journals like Wall Street Journal, Associated Press, Financial Times Limited, Foreign Broadcast Information Service, Los Angeles Times. All these materials collected from the source without any error corrections. Table 1.1 shows the details of the 5 collection disks. The TREC topics have been divided into files based on the TREC task they were used in. Relevance judgments for these topics against various portions of the TREC collections are available.

Relevance judgments. or the right answers, are a vital part of a test collection. TREC uses the following working definition of relevance: If you were writing a report on the subject of the topic and would use the information contained in the document in the report, then the document is relevant. Only binary judgments ("relevant" or "not relevant") are made. and a document is judged relevant if any piece of it is relevant (regardless of how small the piece is in relation to the rest of the document).

Disk	#documents	Size
1	284,550	1036 MB
2	336,310	1126 MB
3	231,219	877 MB
4	260,000	945 MB
5	295,000	1195 MB

Table 1.1: Details of TREC disks.

For this dissertation, the document collections on TREC disks 4 & 5 are used.

Chapter 2

Recent Approaches

TREC-11, organized in 2002, started a new track: *novelty track*. The basic task of this track was: given a TREC topic and an ordered list of relevant documents (ordered by relevance ranking), find the relevant and novel sentences that should be returned to the user from this set. The basic input data for the novelty track was a set of 50 topics taken from TRECs 6, 7, and 8 (topics 300-450). Each topic was tagged with 25 relevant documents. 13 groups participated in this task. Out of those 13 approaches, we will study the 3 most promising approaches.

Two approaches among the three performed expansion of query. This indicates the importance of expansion. Queries are short and contain few key-terms. To get the full information from those short sentences, we need to expand them to retrieve most of the relevant sentences.

2.1 Thesaurus-based Expansion

Tsinghua University[2] participated in TREC 2002 with the basic idea of term expansion. Expansion based information retrieval shows better results at the sentence level. It is possible that a relevant sentence does not match the query if only the original topic or document words are used. Proper expansion of document or query or both is necessary to overcome this limitation. During expansion, certain terms for expansion are selected, and synonyms, hypernyms, hyponyms of those terms are added to the text to be expanded. Expansion techniques can be applied at three different levels during retrieval:

- Query expansion based on thesaurus/statistics feedback
- Replacement-based document expansion
- Combined techniques

2.1.1 Query Expansion

Instead of matching the meaning, if we try to match the words in the topic then we have to reject most of the sentences which should be considered as relevant. Here query expansion (QE) technology is necessary and helpful. Basically, there are two approaches for QE:

- **Using Thesaurus:** Expand terms of the topic using a standard thesaurus, i.e. WordNet, Dekang's thesaurus[3].

- **Local co-occurrence:** Expand using terms that frequently co-occur in a fixed window size with any of headwords in the relevant document set, called *local co-occurrence expansion.*

2.1.2 Document Expansion

Usually it happens that the query mentions a general topic while some relevant documents describe detailed information. For example, the concept of "vehicle" in a query may be expressed by specific words such as "car", "truck" in a document. In this case

- QE may take too many useless words because of aimless expansion

- Setting weights for the original and expanded terms is one of the main difficulties in QE.

In this case, term expansion in documents is helpful. Just like QE, for document expansion (DE) any standard thesaurus can be used.

2.1.3 Combination of QE and DE

QE and DE are oriented from two aspects of retrieval problem but for certain topics the combination works well. Expansion of topics can be classified into either QE or DE according to topic and document characteristics. Two intuitive approaches are:

- **Topic-oriented:** Define different fields' similarities in the topic:
  FS_{td} : similarity between <title> and <desc>
  FS_{tn} : similarity between <title> and <narr>
  FS_{dn} : similarity between <desc> and <narr>

  If $FS_{dn} < \theta_1$ and $(FS_{td} + FS_{dn} - 2FS_{tn}) < \theta_2$ then topic should use DE, otherwise QE is performed. In other words we can define, if <desc> and <narr> fields have less similarity then we have more key-terms. Again if the <title> and <desc> fields are not subsets of the key-terms present in <narr> then we have enough key-terms in these 3 fields. In this case DE should be performed, otherwise QE is used. The thresholds θ_1 and θ_2 are set respectively 0.07 and 0.035 by Tsinghua University in TREC 2002.

- **Document-oriented:** Compute the value of

$$\frac{number\ of\ words\ to\ be\ expanded}{number\ of\ words\ in\ documents}$$

for each topic. If the value is greater than θ, then documents contain enough terms for expansion, perform DE, else use QE. If there are enough terms for expansion with respect to the documents' size then DE should be performed. θ is set to 0.058 in TREC 2002.

2.2 Pseudo-Relevance Feedback

Queens College[4] participated in TREC 2002 with pseudo-relevance feedback approach. They employed all sections of a topic to form long queries for retrieval because the 'documents' are actually short sentences. The queries have, on an average, 19.14 unique terms. In relevance feedback, relevant sentences of initial retrieval are used as feedback information. A new query is formed using these feedback sentences for final retrieval. In pseudo-relevance feedback, top ranked documents of initial retrieval are assumed as relevant and used as feedback for new query construction.

Briefly their IR system can be described by following steps:

- Perform initial retrieval without pseudo-relevance feedback.
- Set *retrieval status value*(RSV) threshold (*tr*) values on the ranked retrieval list to decide the relevance of the retrieved sentences.
- Retrieve sentences with RSV > *tr* are considered relevant.

Queens College employed *tr*=1.25 and 1.5.

2.3 Approach by IRIT-SIG

IRIT[5] participated with a new and simple idea in TREC 2002. They performed certain processing before relevant and novelty retrieval. All topics and sentences are considered as texts for uniformity in processing.

2.3.1 Processing

At first, all texts are pre-processed, then topics and documents are treated separately.

- **Text processing:** Stop words are removed, the remaining words are normalized using a dictionary that provides common roots for inflected words.

- **Topic processing:** Each term in the processed topic is weighted and categorized into 2 groups: highly relevant terms (HT), lowly relevant terms (LT). The following formula is used to compute the term weight:

$$weight(t_i, T_k) = \begin{cases} tf_{i,k} & \text{if } tf_{i,k} \geq 3 \\ 1 & \text{otherwise} \end{cases}$$

  Where, T_k is a topic, t_i a term and $tf_{i,k}$ is the frequency of t_i in T_k. The intuition behind this weighting function is to obtain a significant difference between HT and LT. Each term is categorized into two groups defined as follows:

$$HT_k = \{t_i \mid t_i \in T_k \text{ and } weight(t_i, T_k) > 1\}$$

$$LT_k = \{t_i \mid t_i \in T_k \text{ and } weight(t_i, T_k) = 1\}$$

- **Document processing:** Each term of processed document is associated a weight defined as follows:

$$weight(t_i, S_j) = tf_{i,j}$$

  Where, S_j is a sentence, t_i is a term and $tf_{i,j}$ is the frequency of t_i in S_j.

2.3.2 Relevant retrieval

In order to decide if a sentence is relevant, three components are associated with each sentence:

- A score that reflects the sentence-topic matching:

  Given a topic T_k and a sentence S_j

$$\begin{aligned} Score(S_j, T_k) &= \sum (weight(t_i, S_j) \cdot weight(t_i, T_k)) \\ &= \sum_{t_i \mid t_i \in HT_k} (tf_{i,j} \cdot tf_{i,k}) + \sum_{t_i \mid t_i \in LT_k} tf_{i,j} \end{aligned}$$

- Two groups of terms:

  HS_j corresponds to the highly relevant terms from the topic that occurs in the sentence. LS_j corresponds to the lowly relevant terms from the topic that occurs in the sentence. In mathematical expression,

$$HS_j = \{t_i \mid t_i \in (S_j \cap HT_k)\}$$

$$LS_j = \{t_i \mid t_i \in (S_j \cap LT_k)\}$$

	Relevant
Baseline	0.040
thunv1	0.235
thunv2	0.235
thunv3	0.235
CIIR02tfnew	0.211
thunv4	0.225
CIIR02tfkl	0.211
pircs2N02	0.209
pircs2N01	0.209
pircs2N04	0.197
ss1	0.186

Table 2.1: Average F scores of TREC 2002 novelty track

A given sentence S_j is considered as relevant iff:

$$Score(S_j, T_k) > f\left(\frac{|LS_j|}{|LS_j| + |HS_j|}\right) \cdot |HT_k| + g\left(\frac{|HS_j|}{|LS_j| + |HS_j|}\right) \cdot LT_k|$$

For TREC run, IRIT set $f()$ and $g()$ to:

$$f(x) = \begin{cases} 2 & \text{if } x = 0 \\ 1.5 & \forall x \in (0, 1] \end{cases}$$

$$g(x) = \begin{cases} 0.85 & \text{if } x = 0 \\ 0.3 & \forall x \in (0, 1] \end{cases}$$

2.4 TREC 2002 result summary

Table 2.1 gives the result list of best runs in TREC 2002 novelty track submitted by different participants[6]. For baseline result, random sentences are picked from the collection. All these results are average F scores of the 50 topics.

thunv{1,2,3,4} were the four runs submitted by Tsinghua University. CIIR02tt-{new,kl} were the two runs submitted by University of Massachusetts. pircs2N0-{1,2,4} were submitted by City University of New York. ss1 was submitted by Streamsage.

Chapter 3

Expansion and Feedback

Out of all the approaches described in the previous chapter, expansion and relevance feedback approaches provided better results. Any new technique could be a extension of these two methods. So, to achieve a base level result from those two techniques we performed certain experiments. These two techniques were separately implemented and all retrieval experiments were performed on the TREC 2002 novelty data set and Disk 4 & 5 adhoc document collection. Novelty data set includes 47,620 sentences and 50 topics selected from topic 300-450.

TREC provides relevant documents for relevant setence retrieval, but we tried to retrieve relevant sentences using and without using the relevant document list. Here the idea was to make IR system independent of such a list. So, we provide our result for both cases.

3.1 Baseline Runs

For baseline runs, the following steps were performed:

- all fields of topics were indexed
- sentences of documents were indexed
- simple, similarity based retrieval was done.

For all experiments we selected 2 thresholding schemes and 3 weighting schemes to derive certain conclusions from the result

Present weighting schemes: Following 3 different weighting schemes were used for the runs:

- *lnn*: $tf_{new} = \ln(tf) + 1.0$, no conversion is done for *idf* and document normalization

ω	10.0	11.0	12.0	13.0	14.0	15.0	16.0
lnn	0.142	0.155	0.169	0.175	0.169	0.171	0.166

Table 3.1: Baseline retrieval without relevance filter

- *ltn*: $tf_{new} = \ln(tf) + 1.0$, $weight_{new} = tf_{new} * \log(\frac{number\ of\ docs}{frequency\ of\ term})$, no conversion is done for document normalization
- *ltc*: $tf_{new} = \ln(tf) + 1.0$, $weight_{new} = tf_{new} * \log(\frac{number\ of\ docs}{frequency\ of\ term})$. for weight normalization divide each $weight_{new}$ by $\sqrt{\sum weight_{new}}$

Present thresholding strategies: During final retrievals, we selected 1000 top ranked sentences. Before F score comparison we picked certain number of top ranked sentences from those 1000 sentences by two different thresholding techniques:

1. Pick all sentences having similarity $> \omega$, where ω is absolute similarity.
2. Pick all sentences having similarity $> (\theta *$ highest similarity for that topic), where $0 < \theta < 1$.

By this thresholding technique, we tried to trace a cut-off similarity value independent of the topic/document can be used as standard cut-off for retrieval.

3.1.1 Results and Analysis

Table 3.1 to 3.4 shows the result without relevance filter and absolute similarity(ω) thresholding. Table 3.5 to 3.8 shows the result without relevance filter and θ thresholding.

For Table 3.1 to 3.3, ω is selected by manually checking the result after retrieval. It is not possible to choose ω manually, but this test is used to check the pattern of similarities among topics and to compare the weights. We found all the results first increasing and then decreasing. For Table 3.4, θ is varied so that we can get a stable θ with good performance. Here we found with relevance filter, we get good performance for θ= 0.35 to 0.40.

Table 3.5 to to 3.7, follows the usual pattern of performance with absolute similarity ω. In Table 3.8, retrieval achieved good performance for θ=0.20 to 0.25.

In above both runs, we found weight *lnn* achieved better results in ω thresholding, so ω thresholding is found to be more applicable weight. In our next experiments, retrieval performed on all weights just to check our intuition. We found with relevance filter, we got good performance for θ=0.20. We performed our rest of the experiments with relevance filter, to establish our intuition about θ.

ω	3.0	3.5	4.0	4.5	5.0	5.5	6.0
lnc	0.126	0.140	0.142	0.143	0.132	0.128	0.109

Table 3.2: Baseline retrieval without relevance filter

ω	50.0	60.0	70.0	80.0	90.0	100.0	110.0
ltn	0.149	0.166	0.162	0.170	0.169	0.166	0.145

Table 3.3: Baseline retrieval without relevance filter

θ	0.2	2.5	3.0	3.5	4.0	4.5	5.0
lnn	0.128	0.148	0.166	0.168	0.173	0.165	0.154
lnc	0.113	0.131	0.140	0.140	0.133	0.126	0.111
ltn	0.156	0.168	0.169	0.175	0.166	0.165	0.153

Table 3.4: Baseline retrieval without relevance filter

ω	10.0	11.0	12.0	13.0	14.0	15.0	16.0
lnn	0.210	0.214	0.217	0.219	0.203	0.201	0.190

Table 3.5: Baseline retrieval with relevance filter

ω	0.5	1.0	1.5	2.0	2.5	3.0	3.5
lnc	0.178	0.183	0.190	0.193	0.188	0.187	0.184

Table 3.6: Baseline retrieval with relevance filter

ω	5.0	10.0	20.0	30.0	40.0	50.0	60.0
ltn	0.183	0.189	0.204	0.207	0.200	0.192	0.196

Table 3.7: Baseline retrieval with relevance filter

θ	0.10	0.15	0.20	0.25	0.30	0.35	0.40
lnn	0.195	0.205	0.215	0.217	0.214	0.207	0.205
lnc	0.186	0.201	0.207	0.200	0.193	0.177	0.159
ltn	0.207	0.210	0.213	0.208	0.203	0.201	0.186

Table 3.8: Baseline retrieval with relevance filter

3.2 Thesaurus-based expansion

The novelty track data set is meant for retrieval of new sentences from the given data for a given user query. In other words, one has to remove the redundant information from the data. Novel sentences can be retrieved in two passes. In the first pass, all relevant sentences are selected and in the second pass,the redundant sentences removed. Retrieval in the first pass with good F score is very crucial for the second pass novelty retrieval. In other words, we can say the second pass performance depends on the performance of the first pass. Our intention was to improve the sentence level relevant retrieval for the first pass.

We used Dekang's dependency thesaurus[3] for expansion of the topic. The following steps are performed during expansion process:

- Remove the stop words from topics.
- Index the stemmed words, phrases and unstemmed words.
- Expand the unstemmed words of indexed topics using Dekang's dependency thesaurus.
- Re-index the topics after expansion. Adjust the weight of the indexed words using following function:

$$\begin{aligned} weight'(x) &= f(weight(X), strength(X, X')) \\ &= (\alpha + (1 - \alpha) \cdot strength(X, X')) \cdot weight(X) \end{aligned}$$

Where, X is the word to be expanded,
X' is the new word obtained from thesaurus,
$weight(X)$ gives the initial weight of X,
$strength(X, X')$ returns the dependency strength
between X and X' supplied by thesaurus,
x is the word after stemming X',
$weight'(x)$ is new weight of x,
α is the constant parameter set to 0.2.

The weight of the new word X' to be added to the topic should be a function of weight of X and $strength(X, X')$. Product of $strength(X, X')$ and $weight(X)$ should share a main portion of new weight. We constructed the above formula by adding 20% of weight of original word and 80% of product of $strength(X, X')$ and $weight(X)$.

	P	R	F
Without expansion	0.12	0.62	0.185
With expansion	0.13	0.29	0.152

Table 3.9: Results with and without expansion

```
<num> Number:  322
<title> International Art Crime
<desc> Description:
Isolate instances of fraud or embezzlement in the
international art trade.
<narr> Narrative:
A relevant document is any report that identifies an
instance of fraud or embezzlement in the international
buying or selling of art objects.  Objects include
paintings, jewelry, sculptures and any other valuable works
of art.  Specific instances must be identified for a
document to be relevant; generalities are not relevant.
```

Figure 3.1: Case study: Topic 322

3.2.1 Results

For baseline comparison, we performed a plain retrieval without any expansion. Table 3.1 give the comparison of P, R, F scores of both retrievals. From the table it is clear that precision is increased but the recall is decreased.

3.2.2 Analysis

Result shows that the retrieval after expansion of topic using the thesaurus is not improved. To analyze the case we selected the topic 322. TREC provides 34 relevant sentences for this topic. Without expansion, 14 relevant sentences retrieved. With expansion, only 2 relevant sentences are retrieved. For this topic, the 10^{th} sentence of document LA092989-0086 is relevant. This sentence is selected by baseline retrieval but rejected by the retrieval after expansion. Figure 3.1 and Figure 3.2 show the topic and sentence respectively.

During topic expansion “instance”, “trade”, “international”, “embezzlement”, “isolate”, “crime”, “fraud”, “art” are expanded. During document indexing “touch”, “work”, “deal”, “detect”, “seiz”, “art”, “print”, “raid”, “arrest” etc. terms were indexed. In baseline retrieval, the frequent occurrence of “art” and presence of “work”

```
<NewDocId> LA092989-0086:10
<Text> Gates said Thursday after detectives seized
more than 1,600 art prints and other works in a series
of raids touched off by the arrest of an art dealer
trying to peddle a phony Renoir.
```

Figure 3.2: Case study: 10_{th} sentence of LA092989-0086 document.

accepted the sentence as relevant. But after expansion, the topic is now crowded with terms related to "fraud" and "crime", as the thesaurus provides 154 terms for "crime" and 167 terms for "fraud". In the sentence we don't have any strong word related to "crime" or "fraud". So, this made the retrieval system to drop the similarity factor very low value and eventually the sentence was rejected.

After analyzing the case, we found the expansion on the basis of dependent synonyms is not much helpful. Better result can be achieved if we use the a thesaurus which can provide us the hyponyms or hypernyms of words. Just like "seiz" and "detective" are related to "crime" and "fraud" but they are not synonyms. In expansion based IR systems the thesaurus plays a crucial role.

3.3 Adhoc feedback-based expansion

We tried to expand the topics using novelty track sentences and using document collection d45.

3.3.1 Expansion using sentences

For adhoc expansion, the sentences of the novelty track is used, then the expanded query can be used for relevant sentence retrieval. Following steps describe the process:

- Run the initial topics on novelty track sentences
- Assume the top $N(=20)$ documents to be relevant
- Build feedback query
- Run expanded query on the sentence collection

Table 3.10 to 3.13 show the result for this experiment. For ω thresholding. *lnn* weight performed better than other weights. We achieved better results for θ=0.2 shown in Table 3.13.

In comparison to baseline results, this experiment did not perform well. We can conclude that the sentences don't posses enough relevant terms for a query expansion.

ω	0.3	0.4	0.5	0.6	0.7
lnn	0.205	0.211	0.212	0.208	0.192

Table 3.10: Adhoc on sentences with relevance filter

ω	0.03	0.05	0.07	0.09	0.10
lnc	0.181	0.188	0.194	0.193	0.192

Table 3.11: Adhoc on sentences with relevance filter

3.3.2 Expansion using documents from d45

The novelty data set of TREC 2002 is collected from the relevant documents of disk 4 & 5 (*d*45)[1]. So, *d*45 documents can be used for query expansion. And then the expanded query can be used for relevant sentence retrieval. Following steps describe the process:

- Run the initial topics on *d*45
- Assume the top $N(=20)$ documents to be relevant
- Build feedback query
- Run expanded query on the sentence collection

Table 3.14 to 3.17 show the results for this experiment. Here we can see, *lnn* weight performed better than others in Table 3.14 to 3.16. Form Table 3.17, we can check θ performed better for range 0.15 and 0.30. So, on an average θ=0.20 can be considered as standard cut-off.

In adhoc expansion, the performance is improved with respect to baseline and expansion on sentences. Still, the only d45 expansion is not able to improve the performance.

3.4 Conclusion

The intention of this dissertation was to study the affect of expansion in sentence-level retrieval. We tried with different thresholding schemes and with different weights for

ω	1.0	1.5	2.0	2.5	3.0
ltn	0.205	0.205	0.209	0.205	0.195

Table 3.12: Adhoc on sentences with relevance filter

θ	0.10	0.15	0.20	0.25	0.30	0.35	0.40
lnn	0.192	0.208	0.214	0.207	0.210	0.206	0.197
lnc	0.187	0.201	0.202	0.199	0.193	0.178	0.160
ltn	0.207	0.217	0.211	0.203	0.197	0.193	0.189

Table 3.13: Adhoc on sentences with relevance filter

ω	0.2	0.3	0.4	0.5	0.6	0.8
lnn	0.194	0.200	0.211	0.209	0.204	0.176

Table 3.14: Adhoc on d45 with relevance filter

ω	0.02	0.04	0.06	0.08	0.10	0.12
lnc	0.177	0.179	0.187	0.192	0.191	0.183

Table 3.15: Adhoc on d45 with relevance filter

ω	0.3	0.6	0.9	1.2	1.5	1.8
ltn	0.184	0.195	0.202	0.208	0.203	0.201

Table 3.16: Adhoc on d45 with relevance filter

θ	0.10	0.15	0.20	0.25	0.30	0.35	0.40
lnn	0.190	0.202	0.214	0.213	0.219	0.208	0.206
lnc	0.183	0.194	0.204	0.198	0.194	0.183	0.168
ltn	0.206	0.217	0.214	0.214	0.208	0.202	0.190

Table 3.17: Adhoc on d45 with relevance filter

the retrieval system. After three types of expansion. we can conclude that thesaurus based expansion performance crucially depends upon the thesaurus. If the thesaurus is chosen carefully. then the result can be improved. In adhoc expansion. the adhoc on sentences not performed better. The d45 adhoc expansion though performed a little better but not with a good F score. Among the weights, *lnn* weight based retrieval is better than other weights. With relevance filter. we achieved better performance with $\theta = 0.2$.

Appendix A

Statistical Tests of Significance

A.1 Statistical Testing

Traditional evaluation methods use overall, or average performance measures for an IR system on a set of topics. Comparing two approaches using these average measures is sometimes misleading, since traditional testing methods fail to provide a measure for comparing performance on individual queries. Two systems should be compared query by query to disclose the real efficiency in retrieval[7]. For such reasons statistical testing is more logical. In our experiment, we used statistical testings to compare two retrieval methods, i.e. retrieval methods were compared pair-wise.

Let X_i and Y_i be the scores of retrieval methods X and Y for query i, where $i = 1 \cdots n$ and define

$$D_i = Y_i - X_i$$

We considered three different significance tests for our experiment, the t-test as well as its nonparametric alternatives, the Wilcoxon signed ranks test and the Sign test.

A.1.1 Paired t-test

$$t = \frac{\overline{D}}{s(D_i)/\sqrt{n}}$$

where,

$$\overline{D} = \frac{1}{n}\sum_{i=1}^{n} D_i$$

and

$$s(D_i) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(D_i - \overline{D})^2}$$

Test	t-test	w-test	s-test
Baseline vs. Adhoc on sentences	accepted	accepted	rejected
Adhoc on sentences vs. Adhoc on d45	accepted	accepted	rejected
Baseline vs. Adhoc on d45	accepted	accepted	rejected

Table A.1: Significant test result

A.1.2 Paired Wilcoxon test

$$T = \frac{\sum R_i}{\sqrt{\sum R_i^2}}$$

where, R_i = sign(D_i) $*$ rank$|D_i$

A.1.3 Sign test

Define the statistics s^+ and s^- as the number of queries with $D_i > 0$ and $D_i < 0$ respectively, and let $n = s^+ + s^-$. Any cases where D_i= 0 are ignored. The final test statistics:

$$T = \frac{2s^+ - n}{\sqrt{n}}$$

A.2 Test Runs

For each test we have 50 pairs of data, i.e. $n = 50$. When $n > 30$ all of the above tests follow standard Normal distribution. Table A.1 shows the statistical significant test ran over baseline result, adhoc on sentences and adhoc on d45.The value for the statistics under the null hypothesis H_0: the two runs are equally effective, is accepted by t-test and w-test. As the null hypothesis is accepted. we can conclude that the 3 tests are significantly same. But the s-test rejected all the tests. showing a significant difference in means.

Bibliography

[1] TREC site. http://trec.nist.gov/.

[2] Min Zhang, R. Song, C. Lin, S. Ma, Zhe Jiang, Y. Jin, Y. Liu, and Le Zhao. Expansion-Based Technologies in Finding Relevant and New Information. *TREC Publications*, 2002.

[3] Dekang Lin. Dependency Thesaurus, http://www.cs.ualberta.ca/~lindek/.

[4] K. L. Kwok, P. Deng, N. Dinstl, and M. Chan. TREC 2002 Web. Novelty and Filtering Track Experiments Using PIRCS. *TREC Publications*, 2002.

[5] Taoufiq Dkaki. Josiane Mothe, and Jérôme Augé. Novelty Track at IRIT-SIG. *TREC Publications*, 2002.

[6] Donna Harman. Overview of the TREC 2002 Novelty Track. *TREC Publications*, 2002.

[7] David Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. *ACM SIGIR*, pages 329–338. 1993.