IIT at TREC-10

M. Aljlayl, S. Beitzel, E. Jensen
Information Retrieval Laboratory
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
{aljlayl, beitzel, jensen } @ ir.iit.edu

A. Chowdhury
AOL Inc.
chowdhury@ir.iit.edu

D. Holmes
NCR Corporation
David.Holmes@WashingtonDC.NCR.COM

M. Lee, D. Grossman, O. Frieder
Information Retrieval Laboratory
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
{lee, grossman, frieder} @ ir.iit.edu

Abstract

For TREC-10, we participated in the adhoc and manual web tracks and in both the site-finding and cross-lingual tracks. For the adhoc track, we did extensive calibrations and learned that combining similarity measures yields little improvement. This year, we focused on a single high-performance similarity measure. For site finding, we implemented several algorithms that did well on the data provided for calibration, but poorly on the real dataset. For the cross-lingual track, we calibrated on the monolingual collection, and developed new Arabic stemming algorithms as well as a novel dictionary-based means of cross-lingual retrieval. Our results in this track were quite promising, with seventeen of our queries performing at or above the median.

1 Introduction

For IIT at TREC-10, we focused on the adhoc tasks (both automatic and manual), the site finding task, and the Arabic cross-lingual tasks. For the adhoc tasks, our system is quite different from last year. We calibrated with different fusion approaches and found that a single similarity measure outperformed our other approaches. We also worked with the NetOwl entity tagger to improve our phrase recognition. In the manual track, we developed a new user interface to assist our manual user.

Our results for the Arabic cross-lingual track were quite promising. We developed a new stemmer and made use of a dictionary-based algorithm that requires the translation of the term to be equivalent when going from Arabic-English and from English-Arabic.
Finally, we participated in the web site finding track. We tested a variety of simple approaches, but unfortunately, our results were not very impressive. We are conducting failure analysis on this track to include in the final paper.

2 Adhoc

For TREC-10's ad-hoc task, we focused on effectiveness for short queries. We did a variety of calibrations after TREC-9 on the utility of fusion of various IR approaches. We found that when the stop word lists and parsers are kept constant and effective ranking strategies are used, essentially similar result sets occur for a variety of similarity measures and improvements in average precision due to fusion are negligible. We published this result [7], and for TREC-10, focused on a single similarity measure.

In this section, we briefly describe our query-processing techniques: the use of automatic statistical phrase weighting based on query length and the use of entity tagging for query terms. In the last section, we present our TREC 10 ad-hoc results including some of our results from fusion.

2.1 Query Processing

Many different strategies are used to improve the overall effectiveness of an IR system. Several examples are automatic term weighting [1, 2] and relevance feedback [3]. Phrases are frequently suggested as a means for improving the precision of an IR system. Prior research with phrases has shown that weighting phrases as importantly as terms can cause query drift [5] and a reduction in precision. To reduce query drift, static weighting factors are applied to a phrase reducing the contribution of importance to a documents ranking. These static weighting factors were shown to yield slight improvements in effectiveness [4, 5]. This year we applied two techniques to improve phrase processing. The first is an automatic phrase-weighting algorithm based on the query length and the second is entity tagging using SRA's NetOwl tagger to determine what phrases to use for search.

2.2 Automatic Statistical Phrases Weighting Based on Query Length

Statistical phrases are frequently identified at index time by identifying two term pairs that occur at least X times and do not cross stop words or punctuation. Twenty-five is commonly used as a threshold for the number of documents a phrase must occur in before it is considered a statistical phrase [5].

While the use of phrases is a precision enhancing technique, their naïve usage generally reduces IR effectiveness. When multiple phrases are evaluated for a given query, the likelihood of query drift increases. This drift is caused by phrases overemphasizing a given document that does not contain a breadth of the attributes but only a highly weighted phrase. For an example query of "oil company law suits", the phrases: "oil company", "company law" and "law suits" will overemphasize documents not containing all the terms or phrases and cause nonrelevant documents to receive a higher ranking. This overemphasis causes query drift and the precision of a system decreases. To correct this, we introduce a damping factor of *(exp(-1*delta*queryLength)* and apply it to the actual contribution any phrases can supply to a given document. In Equation 1 the complete weighting for a phrase is given.

$$\sum 1 + \left(\frac{1 + \ln(1 + \ln(\text{tf}))}{(.8 + .2 * (docsize / avgdocsize))} * \exp(-1 * \text{delta} * \text{queryLength}) \right) * nidf * qtf$$

Equation 1: Phrase Ranking Algorithm

Where:

- *tf* = frequency of occurrences of the term in the document
- *qtf* = frequency of occurrences of the term in the query
- *docsize* = document length
- *avgdoclength* = average document length
- *N* = is the number of documents in the collection
- *n* = is the number of documents containing the word
- *nidf* = log(N+1/n)

Our hypothesis is that as the number of phrases increase for a query, the likelihood of query drift due to a highly weighted phrase increases. Thus, by adaptively weighting phrases based on query length, we can improve precision by reducing the likelihood of drift. We ran tuning experiments with the TREC 6, 7 and 8 short (title only) queries. We measured the effectiveness of the various runs with no phrases and phrases with various static weights and dynamic weights.

By keeping the phrase weight set to one (equivalent to the weight given to terms) our average precision is reduced by almost 5%. Other researchers have experienced this same result [4, 5]. By reducing our phrase weight by a factor of .5 and .25 our effectiveness improves. While other groups have chosen a fixed static weight of 0.5, short queries continue to improve to 0.25. Table 1 shows the average precision for phrase weights of 1, .5, and .25. Our adaptive phrase weighting enables us to avoid tuning for phrases. A dynamic weighting based on query length determines the likelihood that the phrase will contribute to the weight. Our dynamic approach yields an improvement of 12% over the statically tuned approach on average for the 150 queries. All IIT runs this year use the given phrase weighting approached described above.

	No Phr	**Pwt - 1**	**Pwt - .5**	**Pwt - .25**	**Pwt - Sig**	**No->.25**	**No->Sig**
T6	22.37%	21.02%	22.59%	23.03%	23.13%	2.95%	3.40%
T7	17.57%	15.51%	16.94%	17.68%	17.73%	0.63%	0.91%
T8	23.85%	24.09%	24.47%	24.58%	24.60%	3.06%	3.14%
Avg	21.26%	20.21%	21.33%	21.76%	21.82%	2.21%	2.48%

Table 1: Phrase Weighting Evaluation Runs (Short Queries)

2.3 Ad-Hoc TREC 10 Experiments

Our overall results for Trec-10 Ad Hoc experiments are summarized in the following chart.

Above Median	At Median,	Below Median
32	1	17

For all queries, we used our new weighted statistical phrase processing. In addition, for indexing, we used a modified porter stemmer and conflation class stemming system. This year's baseline title only experiment was **iit01t**. For our submitted run, we used a modified pivoted document length ranking strategy. We used Rocchio positive feedback using 15 terms from the top 10 documents selected in pass one and each new query term was given a factor of .25. In addition, we used the TREC disks 4-5 for collection enrichment with Rocchio positive feedback of 15 terms from the top 10 documents and a weighting of 0.15. Our run with feedback and collection enrichment is shown in Figure 1 below.

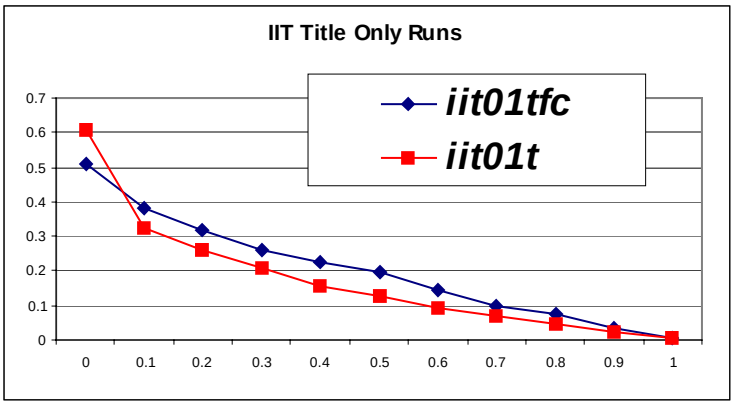

Figure 1: Title only runs

2.4 Query Entity Tagging

We also tested the impact of using an entity tagger over statistical phrases. Tagging a large document collection is difficult with existing entity-taggers because they are not designed for scalability. We were able to tag queries very quickly. The idea was to take the entities tagged in the query and derive two-term phrases from these entities. Hence, a query with "federal housing authority" that has this tagged as a single entity would result in the phrases "federal housing" and "housing authority" to be derived from this tag.

We encountered several problems with this approach. Many queries are not long enough for entity taggers to accurately tag the query terms. Worse, not all queries contain entities that provide useful knowledge of which phrases to use for query processing. To further examine our strategy we used the description of the query instead of only the short titles. Only five of the fifty queries contained entities that were tagged by the NetOwl tagger that could be used for query processing. The five queries and their tags are shown in Table 3. When an entity was encountered, all terms within it were combined as phrases. For query 505 "Edmund Hillary" is identified as a useful phrase, for query 510, "J. Robert Oppenheimer" is found, and for query 527 "Booker T. Washington" is identified as a single phrase. Finally, query 538 has "Federal Housing Authority". Because our index includes only two term phrases, we generate two term phrases from these entities. Future work will focus on tagging the entities in the corpus for indexing. That way, Washington as a name will be distinguished from Washington as a place in both the queries and the index and can be used as a filter.

QUERY 505: WHO IS/WAS <PERSON TYPE="PERSON" FIRSTNAME="EDMUND" LASTNAME="HILLARY" GENDER="MALE">EDMUND HILLARY</PERSON>?
Query 510: Find biographical data on <PERSON TYPE="PERSON" FIRSTNAME="J. ROBERT" LASTNAME="OPPENHEIMER" GENDER="MALE">J. Robert Oppenheimer</PERSON>.
Query 515: What did <PERSON TYPE="PERSON" FIRSTNAME="ALEXANDER GRAHAM" LASTNAME="BELL" GENDER="MALE">Alexander Graham Bell</PERSON> invent?
Query 527: What biographical data is available on <PERSON TYPE="PERSON" FIRSTNAME="BOOKER T." LASTNAME="WASHINGTON" GENDER="MALE">Booker T. Washington</PERSON>?
Query 538: Find documents describing the <ENTITY TYPE="ENTITY" SUBTYPE="GOVERNMENT">Federal Housing Administration</ENTITY> (<ENTITY TYPE="ENTITY" SUBTYPE="GOVERNMENT">FHA</ENTITY>): when and why it was originally established and its current mission.

Table 2: Entity Tagged Queries

2.5 Summary

For TREC-10's ad-hoc task, we focused on effectiveness for short queries for the web track. This year we focused on query processing techniques and fusion approaches. Our initial results are both positive and negative in nature with an overall strong performance in the adhoc title-only task Thirty-two queries of fifty were judged over the median.

3 Manual Task

For the manual WEB Track, IIT expanded upon work from prior years. Our overall results are summarized in the following chart.

Above Median	At Median	Below Median
33	3	14

Research focused on the use of concepts and manual relevance feedback. Additionally, a new user interface was developed. As with previous years, we implemented required and scoring concepts. All fifty topics had at least one required *concept*. A *concept* is represented as a set of words from which a document must contain at least one word. Eighteen topics contained two required concepts (documents must contain at least one entry from each list. Forty-six topics have scoring concepts, or concepts that contribute to relevance but do not identify new documents. Table 3 summarizes our experiments related to concepts. While the use of multiple required concepts only provided a modest boost to average precision, the probability of achieving the best average precision doubled. The median average precision for all teams was 0.1665 for our topics with two required concepts, while the median was 0.1997 for topics where we used one topic, indicating the two concept topics were somewhat more difficult.

Required Concepts	Number of Queries in Set	Avg Precision	Best	At or Above Median, not best	Below Median
1	32	0.3226	7	17	8
2	18	0.3499	8	5	5

Table 3: Average Precision for Manual Queries

We also tested the effect of manual relevance feedback. Manual relevance feedback involved reading some number of documents and selectively modifying queries based upon what was read. To do this, we split the topics into three groups. For the most "top" group, we read at least 100 documents per topic, with a maximum of 156. For the middle group we read between 50 and 99 documents. Finally, we read from zero to 49 documents for the group with minimal relevance feedback. We reviewed a little under 10% of returned documents. Table 4 summarizes the results for manual relevance feedback. It can be seen that reading numerous documents had an impact on whether or not we had the best query.

Documents Read	Number of Queries in Set	Avg Precision	Best Avg Precision	At or Above Median, not best	Below Median
100+	10	0.4714	7	2	1
50-99	25	0.3187	4	15	6
0-49	15	0.2628	4	5	6

Table 4 Manual Relevance Feedback Results

Final results were re-ranked based upon user assessment. User assessed "Relevant" documents contained all elements of topic, "Probably Relevant" contained most elements, or loosely addressed all elements. Documents assessed "Probably not relevant" contained some reference to the topic but did not seem related, while "Not Relevant" were completely unrelated. Table 5 below shows our in-house assessments of the result documents.

User Assessment	**Documents**	**Ranking Adjustment**
Relevant	598	Ranked above all other documents returned
Probably Relevant	523	Relevance score boosted by 0.25
Probably not Relevant	612	Relevance score lowered by 0.5
Not Relevant	1678	Relevance score lowered by 0.9

Table 5 Relevance Assessments from our Manual User

4 Homepage Finding

This year our group participated in the new site finding task. For a baseline run, we indexed the title terms from the document collection and ran an initial query pass using our basic adhoc retrieval strategy. In addition, the source URL's for each result document were cleaned to remove extraneous words and characters so they would adhere to a typical URL format. After having retrieved the results from our initial query pass, we used three techniques to augment and improve the result set: TAND, Co-occurrence Boosting, URL-folding.

4.1 TAND Initial Results at Thirty Percent

The results from the initial query pass were TAND'ed. In order for a candidate result document to remain in the result set, it had to contain a minimum of thirty percent of the query terms. This technique was used as a coarse-grained filter, eliminating result documents that had little chance of being relevant. We arrived at thirty percent and all other thresholds by calibrating with the training site-finding set.

4.2 Boosting on Result Co-Occurrence

Along with our primary title-only index, we created several other indexes that were used for a form of collection enrichment. These included:

- ODP Descriptions – We crawled the hierarchy of the Open Directory Project (www.dmoz.org) and created an index of the description terms for each entry.
- ODP Anchor Text – An index of the anchor text used for hyperlinks in the Open Directory Project
- First-100 – An index of the first one hundred terms from each document in the WT10G.

After the TAND'ing of the result sets from the initial query pass was complete, we ran a query pass against each of these three indexes, and used the following algorithm to "boost" results in the initial result set:

- For the top thirty results from the ODP description query, we checked the URL for the result document in question against the result set from our initial query pass.
  - If it was present in the initial query pass, the score for the document in the initial result set was increased by 85%
  - If it was not present in the initial query pass, but a document with the same URL was confirmed to exist in the WT10g collection, that document was added to the initial result set with the unmodified weight from the ODP Description result set.
- This process was repeated for the two additional indexes in the following order, with the following parameters:
  - ODP Anchor Text: Examined the top sixty results and boosted matches by 50%
  - First-100: Examined the top sixty results and boosted matches by 60%

TAND'ing and Boosting improved our baseline mean reciprocal rank by approximately 70%. It should be noted that the order in which the boosting indexes were queried is very important, as potential results could have been boosted multiple times depending on which source located them first.

The order in which the boosting indexes were queried, and the various boosting factors and number of results examined were determined experimentally by performing a large number of calibrations using the supplied training data for the Homepage finding task. Essentially, the

numbers describe the measure of confidence we placed in the ability of each source to yield relevant results. We found that the ODP indexes, potentially due to the large amount of human oversight and interaction, were trustworthy. By contrast, the index of the first one hundred terms was shown to be less likely to contain highly relevant results, probably due to the presence of large quantities of "noise" information that is often present in the first terms of a web page, such as advertisements, etc.

4.3 Folding

The final technique we used on the boosted result set was our URL-folding algorithm. The idea here is to combine results from the same site in the ranked list so as to order them in a reasonable way. We refer to pages on a web site in terms of *parent-child* relationships. A parent page is shallower in the site hierarchy (e.g.; ir.iit.edu) while a child page is deeper (e.g.; ir.iit.edu/researchers). Folding took place as follows:

a. Parent occurs higher in the result set than child: child is removed from result set and parent's score is increased
b. Child occurs higher than parent: parent score is increased, but child is left in its original position of the result set.

Relevance score modifications were performed for each parent according to the following equation:

$$S_p = S_p + \ln\left(\sum S_c\right)$$

Equation 2: Parent Weight Incrementation

After experimenting with this scheme, we found a paradox: Many parent pages had too many children above them in the rankings, but increasing the increments by which parents were weighted caused parents with many children to be ranked too highly. To provide finer-grained tuning of how parents had their ranks increased, we added a final step to our algorithm that occurred after all folding had been completed. In this step, we moved parent pages that had unfolded child pages of within 35% of the parent's score just above those unfolded children in the result ranking. We also guaranteed that parent pages had at most three unfolded children above them in the ranking, regardless of their relevance.

After attending TREC, we performed some failure analysis on our techniques, in an effort to discover why there was such a large disparity between our performance on the training queries, and our performance on the supplied topics. This failure analysis revealed some deficiencies in our query and document parsers, and also confirmed that there is a high degree of overlap in the improvements observed from our boosting and folding techniques.

Our experimental results for both the training data and the actual homepage topics for each approach are shown in Table 6. The improvements resulting from our post-conference failure analysis are also included. All values express the mean reciprocal rank over the query set.

Query Set	Baseline	Baseline + Boosting	Baseline + Folding (iit01st)	Baseline + Boost + Fold (iit01stb)
Training Data	.590	.725	.670	.880
Homepage Topics	.253	.503	.559	.578
Topics - Improved	.373	.519	.561	.664

Table 6 Results of Site Finding Task (MRR)

5 Arabic Monolingual and Cross-lingual Track

For the Cross-Lingual Arabic Information retrieval, our automatic effort concentrated on the two categories; English-Arabic Cross-Language Information Retrieval (CLIR) and monolingual information retrieval. For the English-Arabic CLIR we used two types of dictionary-based query translation: Machine-Readable Dictionary (MRD) and Machine Translation (MT). The First-Match (FM) technique is used for term selection from a given entry in the MRD [8].

5.1 Monolingual

For the monolingual run, we used two stemming algorithms. The first algorithm is root-based, and second is light stemming. In the root-based algorithm, the main aim is to detect the root of the given word. When no root is detected, the algorithm retains the given word intact. The root-based algorithm is aggressive. For example, the root of *office*, *library*, *book*, and *write* is the same, thus, the root-based algorithm places these in the same conflation class. Accordingly, a light-stemming algorithm is developed. It is not as aggressive as the root-based algorithm. The idea of this technique is to strip out the most common affixes to the Arabic words. For example, it returns the plural, dual to their singular form except for irregular pluralization.

Our monolingual run is described in Table 7. This run did reasonably well, with 21 queries above the median, 1 at the median and three below.

Average Precision								
Best	**Median**	**Worst**	**iit01mlr**	**Above**	**At**	**Below**	**Best**	**Worse**
0.5118	0.2516	0.0216	0.4288	21	1	3	3	0

Table 7 Monolingual run Using Light Stemming

5.2 English-Arabic Cross-Language information Retrieval

We conducted our experiments by using two approaches for query translation. The first approach is the Machine-Readable Dictionary (MRD). The second approach is Machine Translation (MT). In MRD realm, we use the first match in the bilingual dictionary as the candidate translation of the source query term. This approach ignores many noise terms introduced by the MRD. Al-Mawrid English-Arabic is used for the translation process [9].

In MT realm, the translation was performed on every field of the topic individually. We performed our experiment by using a commercial MT system product. It is called Al-Mutarjim Al-Arabey. It is developed by ATA Software Technology Ltd [10]. The post-translation expansion technique is used to de-emphasize the extraneous terms that are introduced to the source query after translation.

Our cross-lingual run is described in Table 8. Our run has 17 queries above the median, zero at the median and eight below. There are 3 queries where our run is the best.

Average Precision								
Best	**Median**	**Worst**	iit01xma	**Above**	**At**	**Below**	**Best**	**Worse**
0.5623	0.1701	0.0001	0.3119	17	0	8	3	0

Table 8 CLIR result using Mutarjim Al-Arabey MT system

REFERENCES

[1] Salton G., C. Yang and A. Wong. "A vector-space model for information retrieval", *Comm. of the ACM*, 18, 1975.

[2] A. Singhal, C. Buckley, and M. Mitra, "Pivoted Document Length Normalization", ACM-SIGIR *Conference on Research and Development in Information Retrieval*, 1996.

[3] J. Rocchio. "Relevance Feedback in Information Retrieval. *Smart System - Experiments in Automatic Document Processing*", pages 313--323. Prentice Hall, Englewood Cliffs, NJ, 1971.

[4] A. Turpin and A. Moffat. "Statistical Phrases for Vector-Space Information Retrieval", ACM-SIGIR, 1999: 309-310.

[5] M. Mitra, C. Buckley, A. Singhal, and C. Cardie. "An Analysis of Statistical and Syntactic Phrases", *Fifth RIAO Conference, Computer-Assisted Information Searching On the Internet*, 1997.

[6] TREC at NIST (National Institute of Standards and Technology) trec.nist.gov

[7] A. Chowdhury, D. Grossman, O. Frieder, C. McCabe, "Analyses of Multiple-Evidence Combinations for Retrieval Strategies", ACM-SIGIR, September 2001.

[8] M. Aljlayl and O. Frieder. “Effective Arabic-English Cross-Language Information Retrieval via Machine Readable Dictionaries and Machine Translation,” *ACM Tenth Conference on Information and Knowledge Managemen (CIKM)*, Atlanta, Georgia, November 2001.

[9] Dar El-Ilm Lilmalayin, http://www.malayin.com/

[10] http://www.atasoft.com/products/mutarjim_v2.htm